Proyecto Final de Carrera

Migración de Microsoft Operations Manager a Nagios

Autor: Víctor Navarro Bo
Director: Jose Luis Poza Luján
Viernes 16 de junio de 2011
Ingeniería Técnica en Informática de Sistemas

Índice

Prólogo

La actual situación económica obliga a replantearse los términos en los que se basa cualquier presupuesto. Los recortes no escapan a ningún sector, y el sector de las tecnologías de la información no es una excepción. Estos recortes obligan a los clientes a estudiar alternativas, a barajar posibilidades que permitan reducir los costes a cifras razonables; sin conocimiento alguno hemos activado la maquinaria para el software libre.

El interés o la necesidad que faltaba para implementar soluciones económicas ha llegado en una situación para muchos angustiosa. El momento que para una gran mayoría resulta casi catastrófico, para otro grupo apunta a la posibilidad de abrir un gran mercado de oportunidades.

Una de las filosofías fundamentales del software libre dentro del sector de las tecnologías de la información es pagar por el soporte y/o los conocimientos del producto, y no tanto por el producto en cuestión. Esto obliga a replantear a su vez la filosofía de la empresa, que debe abandonar en parte la capacidad de implementación para focalizarse sobretodo en la capacidad de soporte del producto y formación del propio cliente.

Ante nuevas líneas de soporte y productos relativamente nuevos, el uso de sistemas de monitorización recobra una vital importancia en el negocio para poder prestar el servicio contratado. La detección a tiempo de problemas consigue mantener la calidad del servicio y mejorar el funcionamiento del centro de soporte. Por otra parte el cliente exige disponer de datos que le ayuden a conocer el estado real del servicio.

Nos enfrentamos en este documento ante dos cambios de filosofías; el primer cambio en cuanto a la filosofía tecnológica, que pasa de ser de un producto cerrado con altas prestaciones de serie a un producto abierto con menos prestaciones *innatas*, y en segundo lugar un cambio en la filosofía de trabajo, con nuevas métricas que ayuden al cliente a supervisar si los niveles contratados se cumplen dentro de los términos establecidos.

Introducción

Entorno del proyecto

Dentro del área de soporte de una empresa de tecnologías de la información las herramientas de monitorización son un factor clave para el éxito del negocio. Los sistemas de monitorización son herramientas útiles para la organización TI. La percepción de su utilidad para el cliente viene implícita en la calidad del servicio y en los informes y reportes extraídos de la misma.

Un sistema de monitorización correctamente *parametrizado* brinda a los gestores información relevante sobre la situación de los servicios ofrecidos, que a su vez pueden utilizar dicha información para tomar acciones reactivas; solucionando problemas complejos antes de que provoquen problemas de disponibilidad y/o capacidad, y también tomar acciones proactivas; que mediante el estudio de tendencias y eventos promuevan cambios en la infraestructura con el fin de evitar futuros fallos o carencias tecnológicas.

Actualmente el cliente objeto de estudio implementa un sistema de monitorización basado en la tecnología *Microsoft Operations Manager* (en adelante MOM). Este sistema es el encargado de monitorizar las demás tecnologías que comprenden el contrato, muchas de ellas también tecnologías de Microsoft:

Tecnología	Versión	Servicio que ofrece
Active Directory	Server 2003	Servicio de directorio
Distributed File System	Server 2003	Servicio de documentos
System Center Configuration Manager	SCCM 2007	Servicio de gestión del parque
Microsoft Exchange	Exchange 2003	Servicio de correo
Windows Sharepoint	WSS 3.0	Servicio de colaboración
McAfee	McAfee 8.7	Servicio de seguridad
Microsoft Operations Manager	MOM 2005	Servicio de monitorización

Tabla 1 – Tecnologías implantadas en el cliente estudio

En un planteamiento reciente de renovación tecnológica, se pretende migrar varios servicios a alternativas libres. El objetivo principal es aprovechar la oferta de herramientas libres del mercado, realizando implementaciones de dichas herramientas de forma específica para el cliente.

La herramienta de monitorización deber ser capaz de implementar mecanismos de control tanto en las nuevas tecnologías de código abierto que se desean desplegar como de las tecnologías que anteriormente se monitorizaban con MOM.

Como la propia filosofía del código abierto apunta, será necesario que las nuevas herramientas desarrolladas se publiquen también bajo la licencia GPL, por lo menos, respecto a la publicación de la parte de código libre utilizado. Esto permitirá a la empresa de TI reutilizar el esfuerzo de desarrollo y

minimizar el mantenimiento de las aplicaciones en otros clientes, y al cliente utilizar ese mismo producto en otras áreas donde no interviene la empresa TI.

La planificación de la migración de la infraestructura a tecnologías *Open Source* se extiende durante varios años. La primera línea de actuación afecta a las siguientes tecnologías.

Tecnología actual	Tecnología propuesta	Información
System Center Configuration Manager	SCM	Producto implementado para el cliente basado en el OCS Inventory. Publicado con licencia GPL bajo petición del cliente.
Microsoft Exchange	Zimbra	Versión 'de pago' del sistema de correo Zimbra para sustituir las características prestadas por Exchange.
Microsoft Operations Manager	SSM	Producto implementado para el cliente basado en Nagios. Publicado con licencia GPL bajo petición del cliente.

Tabla 2 – Relación entre la tecnología actual y la objetivo

El objetivo del presente documento es estudiar los mecanismos necesarios para realizar el cambio tecnológico desde MOM a SSM. A lo largo del documento estudiaremos las características de ambos entornos, sus prestaciones y contraprestaciones, que llevarán más tarde a proponer un mecanismo de migración sin riesgos.

Otro de los acuerdos alcanzados con el cliente consisten en transformar el modelo de negocio a las buenas prácticas propuestas por el marco de trabajo ITIL v3. Este marco de trabajo tiene como principal objetivo proponer una métrica clara con la que el cliente pueda comprobar que el servicio ofrecido se satisface dentro de los niveles de servicio contratados.

El sistema de monitorización se ve afectado por algunas de las tareas propuestas por este modelo, por lo que será necesario tener en cuenta dichas definiciones a la hora de *parametrizar* los indicadores y reportes oportunos.

Motivación

El cliente propuso a estudio la revisión de la mayor parte de la tecnología que implementan los servicios contratados. La justificación principalmente es económica, ya que tras los años transcurridos de servicio, el parque

tecnológico ha crecido de forma considerable, lo que llevará a un aumento sustancial del coste en licencias en una futura renovación con Microsoft.

En los tiempos que corren se pide recortar gastos, asumir niveles razonables y buscar propuestas alternativas que ayuden a reducir el presupuesto. El área informática muchas veces es una de las que mayor esfuerzo económico requiere, y por desgracia una de las que menos es capaz de transmitir a la dirección del cliente y al conjunto de la empresa la necesidad de perpetuar dicho gasto.

Es aquí donde entra como alternativa el software libre. Se considera por tanto que el desarrollo de productos dentro de las licencias GPL ha alcanzado un grado de madurez alto, quizás no para sustituir en el mismo nivel a productos como los de Microsoft, pero si como para implementarlos y contribuir en el desarrollo para su sustitución.

El cliente en este aspecto tiene una idea clara sobre la naturaleza del mundo del software libre y las licencias GPL; conoce la diferencia entre gratuidad y libre. Es por ello que en el caso de productos como el SSM (Servicio de monitorización que nos ocupa) o el SCM (Servicio de aplicaciones e inventario) han tenido que costear el desarrollo de las aplicaciones y la formación para su uso.

En el caso de Zimbra incluso ha sido necesario presupuestar una partida para licencias de usuarios, ya que la versión implementada no es la comunitaria debido a las exigencias del cliente. Sin embargo, las licencias en este caso son vitalicias y no están sujetas a ningún tipo de recargo por renovación.

Por otra parte, el cliente es consciente que los términos GPL obligan a la publicación del código libre usado en el desarrollo de las herramientas. Es por ello que su intención es publicar íntegramente dichas herramientas bajo la misma licencia.

A nivel particular, la herramienta de monitorización se ve afectada por todos estos cambios. Es necesario implementar una solución específica para el cliente que se base en Nagios además de una interfaz gráfica. Tras esto, será necesario migrar el actual sistema a la nueva herramienta implementada.

Además de este cambio, será necesario controlar los demás cambios de herramientas que se avecinan en el cliente. Es necesario que los mecanismos de monitorización funcionen correctamente desde un primer momento con las nuevas tecnologías, ya que existe un mayor riesgo de perdida de servicio al tratarse de tecnologías no lo suficiente explotadas ni conocidas hasta la fecha.

Otra modernización importante que ha asumido el cliente en el nuevo contrato es la implantación de una filosofía de trabajo con referencias, que ayude a mejorar los niveles de calidad ofrecidos a los usuarios.

En este punto el cliente opta por las prácticas ITIL, orientadas específicamente a la gestión de tecnologías de la información, y con un amplio *feedback* positivo en el sector. La motivación tanto del cliente como de la organización TI es clara; mejorar la calidad del servicio.

La gestión de los distintos procesos ayudarán por una parte al cliente a conocer de forma más tangible la situación del contrato, y por otra parte ayudará a la organización TI a administrar y gestionar de forma más eficiente los servicios que ofrece.

Tras las distintas fases de diseño, transición y operación del servicio podemos disponer de una maquinaria de gestión bien engranada que se refleje de forma muy positiva sobre los usuarios. Cabe mencionar, además, que el modelo aplicado se toma como piloto y referente para aplicarlo en un futuro en otras áreas.

Este último punto impacta sobre el servicio de monitorización, ya que la herramienta debe ser capaz de ofrecer datos veraces sobre el funcionamiento de la infraestructura.

Objetivos

Tras este estudio y tras realizar el protocolo de migración los dos sistemas de monitorización deben detectar por igual las alertas, eventos y en general el estado de todos los servicios de la infraestructura.

El objetivo de este proyecto es por tanto realizar un estudio que nos ayude a conocer el funcionamiento de ambos sistemas de monitorización, destacando las carencias y ventajas de cada uno de ellos con el objetivo de resolver de una forma u otra todas las posibles situaciones.

Debido a que las herramientas *Open Source* por lo general no tienen la misma longevidad que los productos que actualmente implementa el cliente, resulta de vital importancia que este estudio analice todos los detalles disponibles con el objetivo de explotar al máximo las prestaciones de las tecnologías.

Tras este estudio, se pretende desarrollar un protocolo de migración seguro entre sistemas MOM y sistemas Nagios, basado en el estudio previo y que permita realizar la transición entre ambos sistemas de forma segura y eficiente.

La migración entre ambos sistemas no implica la sustitución completa del anterior, ya que se prevé un periodo de convivencia entre ellos. Esta situación ayudará a depurar el sistema basado en Nagios durante este periodo. La corrección se debe realizar en base a las necesidades del negocio sin que resulte obligatorio replicar toda la información que detecta MOM en la actualidad.

Al estudio de sistemas y a la propuesta de migración se añade otro objetivo marcado por las buenas prácticas ITIL v3; la monitorización de la Disponibilidad y la Capacidad de los servicios.

La herramienta debe poseer los mecanismos oportunos para extraer reportes e información alineada a las métricas sobre niveles de servicio. Esto por lo general exige presentar en forma porcentual la disponibilidad y la capacidad de los servicios.

Estas exigencias obligarán al desarrollo de otras métricas internas que determinen, por ejemplo, el tiempo de reacción ante un fallo de disponibilidad, el tiempo de resolución…

Descripción del documento

Tras conocer el marco del proyecto, la motivación y los objetivos, se pretende en este punto presentar el modelo de documento actual con el fin de facilitar al lector el conocimiento de su estructura.

En primer lugar se pretende realizar un estudio teórico tanto de las tecnologías implicadas como de la metodología ITIL. La idea principal de este punto teórico es situar el campo y ámbito de este escrito, y ofrecer un mecanismo de definición.

Más adelante se realiza el análisis de los sistemas actual y objetivo. En este punto se pretende analizar el funcionamiento de ambos sistemas partiendo de las definiciones propuestas en el primer apartado. Este estudio va dirigido a conocer el funcionamiento e implementación interna de los mismos.

Tras el estudio de sistemas, se expone el desarrollo del protocolo de migración seguro, además de la definición de pruebas de control de funcionamiento del sistema. En esta parte se abordará la fase de transición entre ambos sistemas, en base al estudio anterior realizado.

Estudio teórico

Introducción

En este apartado se pretende realizar un estudio teórico previo para conocer el funcionamiento de las tecnologías implicadas en la migración de sistemas, y las filosofías de trabajo empleadas en la *parametrización* de las herramientas.

El estudio teórico tiene como objetivo marcar un nivel de entendimiento necesario para comprender la totalidad de este documento, centrándose en las características que intervienen en el desarrollo del mismo.

Resulta importante comprender el funcionamiento de un sistema de código cerrado como *Microsoft Operations Manager*. Pese a la limitación propia de un sistema cerrado, se ha conseguido estudiar la implementación del sistema con el fin de conocer su funcionamiento.

Esto se ha efectuado analizando la información de las bases de datos de la aplicación, así como los distintos *Management Packs*[1] instalados en la herramienta.

El entorno de monitorización de Nagios posee una documentación extensa dada su naturaleza libre. El estudio de Nagios se ha realizado mediante la documentación oficial de la aplicación. Este estudio se inicia tras el conocimiento de la herramienta MOM, con el fin de enfocar la migración a las necesidades reales de la infraestructura.

También se añade dentro de este proceso el estudio de la interfaz gráfica que se empleará para la gestión y consulta de Nagios, cuyo nombre del proyecto *Open Source* es Ninja. La herramienta Ninja basa su funcionamiento en la representación gráfica de los objetos y resultados de Nagios, guardados previamente en bases de datos de mysql.

Intentaremos también introducir de forma teórica la metodología de trabajo propuesta por las buenas prácticas de ITIL v3. Los procesos que afectan a la migración del sistema de monitorización son la Gestión de la Capacidad y la Gestión de la Disponibilidad.

La Gestión de la Capacidad se ocupa de estudiar si los servicios ofrecidos están alineados a las necesidades de procesamiento y almacenaje que necesita el cliente. Los servicios no deben ofrecer una capacidad inferior a la necesaria; ya que se provocarían problemas de disponibilidad del servicio, ni tampoco capacidades excesivamente desproporcionadas; lo que se traduce en costes no alineados con las necesidades del cliente.

En cuanto a la Gestión de la Disponibilidad, este proceso se encarga de estudiar en qué porcentaje los servicios están plenamente operativos para los usuarios. El uso de tecnologías de redundancia y alta disponibilidad deben promover métricas capaces de comprobar si, bajo situaciones críticas, los

---

[1] Conjunto de definiciones de configuraciones de monitorización que hacen referencia a una aplicación o servicio.

servicios prestados seguirían en funcionamiento o en que grado estos se verían afectados.

Microsoft Operations Manager

MOM es la herramienta de administración de sistemas de la compañía Microsoft. La herramienta actualmente se conoce como *System Center Operations* Manager, en su última versión de 2007. En el caso que nos ocupa, la versión implementada en el cliente es la de 2005.

MOM ofrece de serie pocas opciones de monitorización de otros productos del mercado. La idea principal es proveer de un centro preparado para la implementación de otros paquetes de definiciones que ayuden a monitorizar servicios específicos (*Management Packs*).

Es por ello que MOM se encarga, entre otras cosas, de la infraestructura de despliegue de agentes de monitorización, de la administración de los paquetes de definiciones y de la visualización del estado de los componentes monitorizados.

Microsoft ofrece a los desarrolladores una amplia documentación para el desarrollo de paquetes de monitorización. De este modo, un fabricante en particular que conoce su producto puede definir reglas y eventos de monitorización, que posteriormente pueden ser aplicados sobre grupos de equipos.

Los agentes distribuidos comprueban la pertenencia o no de los equipos a los distintos grupos. Si se debe monitorizar una determinada tecnología en un determinado equipo, el agente pide al servidor de monitorización MOM la información necesaria proporcionada por los paquetes, para que sea el agente quien reporte a MOM dicha información.

Como resulta obvio, Microsoft es el primer fabricante en proporcionar una larga lista de paquetes de definiciones para la monitorización de sus productos. La totalidad de los servicios ofrecidos al cliente en la actualidad posee un paquete de monitorización proporcionado por el fabricante.

El manejo de estos Management Packs, y del resto de configuraciones de monitorización que la herramienta ofrece, se realiza desde la aplicación *Administrador Console*. Esta aplicación brinda una estructura jerarquizada similar a la del resto de aplicaciones de Microsoft, donde es posible configurar tanto las reglas de monitorización, como los equipos y sus agentes, entre otras cosas.

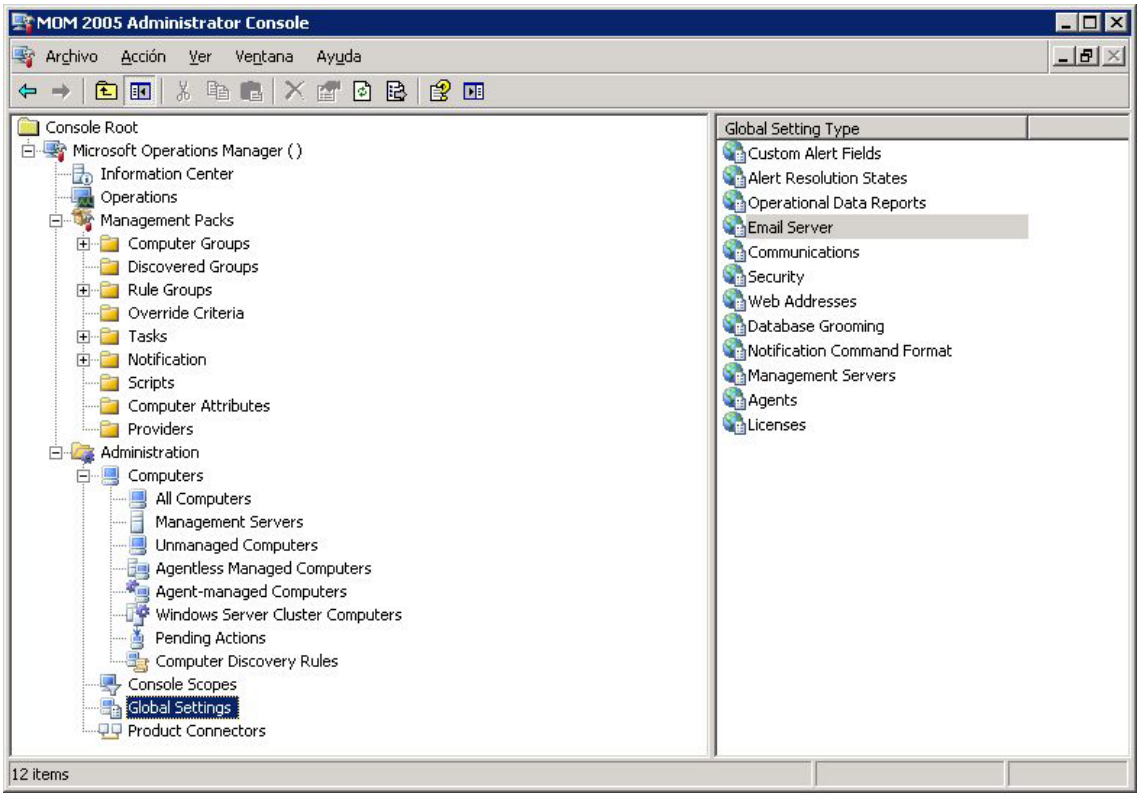

Ilustración 1 – Administrator Console de MOM

Dentro de la *Administrator Console* podemos diferenciar claramente entre las configuraciones cargadas a través de Management Packs, y las configuraciones propias de MOM.

Las configuraciones que se permiten importar a través de un Management Pack son las siguientes:

- **Computer Groups**: definiciones de grupos de equipos según las necesidades de las reglas instaladas: Windows Servers 2003 Systems, Windows Servers 2003 Active Directory…
- **Discovered Groups**: grupos detectados en la infraestructura, en base al descubrimiento de equipos pertenecientes a los mismos que realiza los agentes.
- **Rule Groups**: grupos de reglas orientadas a un servicio. Los grupos de reglas se suelen aplicar sobre grupos de equipos previamente definidos.
- **Override Criteria**: criterios de remplazo de eventos no asociados a reglas. Permite "trivializar" el reporte de eventos y alertas para impedir la saturación.
- **Tasks**: tareas del propio servidor. Monitorización desde el lado servidor, en lugar de por agente distribuido.
- **Notification**: operadores y grupos de operadores que deben recibir las alertas bajo los criterios definidos en los eventos.
- **Scripts**: scripts cargados en el sistema y que pueden ser usados para la monitorización. Habitualmente escritos en Visual Basic Script[2].

[2] Lenguaje de programación interpretado por Windows Scripting Host

- **Computer Attributes**: atributos de equipos a monitorizar por eventos o contadores.
- **Providers**: tipos de fuentes de información para MOM (Event Logs, scripts, contadores…).

Las configuraciones del propio MOM permiten gestionar el parque y configurar la aplicación en sí para su funcionamiento. Estos son sus ítems:

- **Computers**: gestión del parque a monitorizar. Permite distinguir entre los distintos casos de equipos en la infraestructura:
  - **Management Servers**: servidores de monitorización. Generalmente se utiliza un único servidor, pero puede implementarse un modelo en jerarquía.
  - **Unmanaged Computer**: servidores detectados en la infraestructura que no se monitorizan actualmente.
  - **Agentless managed Computers**: equipos sin agente.
  - **Agent-managed Computers**: equipos con agente.
  - **Computer Discovery Rules**: reglas para inspeccionar y encontrar nuevos servidores/equipos.
- **Console Scopes**: permite definir el ámbito de los operadores del sistema.
- **Global Settings**: configuraciones de la aplicación, como por ejemplo, servidor de correo, crecimiento de las bases de datos, agentes…
- **Product Connectors**: permite definir conexiones entre distintos paquetes de monitorización; dependencias, alertas…

Administrator Console es por tanto un área de configuración a todos los niveles de la aplicación. Existe otra interfaz encargada de plasmar los resultados de monitorización en base a la configuración definida anteriormente, a esta interfaz se accede a través de *Operator Console*.

Operator Console permite visualizar los resultados de monitorización bajo una serie de "vistas" que propone la aplicación. Según lo que deseemos supervisar, podemos seleccionar entre:

- **Alerts**: esta vista de forma resumida o extendida las alertas detectadas en las distintas reglas monitorizadas.
- **State**: muestra una relación entre las alertas y los equipos monitorizados, es decir, el estado de los servidores en base a las alertas recibidas.
- **Events**: muestra los eventos detectados y el resultado de las tareas que se ejecutan o bien en el lado cliente o bien en el servidor.
- **Performance**: permite visualizar datos contabilizados a lo largo de un periodo de tiempo, para conocer la evolución del estado del servidor.
- **Diagram**: visualiza un diagrama con la topología de la infraestructura; clústers, nodos, conexiones entre servidores…

Nagios + Ninja

Nagios es un sistema de monitorización de código abierto publicado bajo la licencia GPL. El funcionamiento genérico de Nagios consiste en la

configuración de ciertos servicios en el lado servidor que cada cierto periodo de tiempo activan consultas remotas en los sistemas a monitorizar.

Nagios posee una estructura muy específica basada en la definición de objetos, con una extensa lista de tipos, que pueden definirse de forma jerárquica y relacional. Los distintos tipos de objetos poseen una serie de parámetros que marcan la configuración y el comportamiento de los mismos.

Algunos de los objetos más importantes de Nagios, y que utilizaremos en este estudio son:

- **Host**: referente a un componente de red, ya sea servidores, routers, switches…
- **Hostgroup**: grupo de hosts en base a unos criterios.
- **Service**: evento de monitorización que provoca la ejecución de un comando cada cierto tiempo y que devuelve una salida formateada.
- **Servicegroup**: grupo de servicios en base a unos criterios.
- **Contact**: contacto o usuario que proporciona información para el envío de alertas.
- **Contactgroup**: grupo de contacto en base a unos criterios.
- **Timeperiods**: periodos de monitorización en base a los horarios del negocio.
- **Command**: objeto a ejecutar por un *service* que lanza el binario relacionado.

Como hemos mencionado, los parámetros permiten referenciar los distintos objetos entre si. Así, por ejemplo, es posible aplicar *services* sobre *hosts*, o sobre *hostsgroups*, *contacts* sobre *services*…

Por otra parte la definición jerarquizada permite definir elementos comunes para distintos objetos, con la intención de utilizar como plantilla dichas definiciones para la creación de nuevos objetos. En este caso, si consideramos por ejemplo, que todos los *hosts* compartirán el *contactgroup* "administradores", podemos definir un objeto previo con esta configuración, y posteriormente referenciar los siguientes objetos mediante el parámetro *use*.

Bajo estas premisas es posible configurar distintos eventos de monitorización, tanto en el lado servidor como en el cliente, distintos mecanismos de alertas…

Nagios permite monitorizar de serie ciertos aspectos tanto del sistema local como de sistemas remotos desde el lado servidor mediante la ejecución de servicios basados en consultas SNMP[3]. Esto permite configurar, por ejemplo, servicios para la monitorización del hardware del equipo, del sistema base, o del estado de los servicios ofrecidos en determinados puertos, entre otros.

Sin embargo, para la monitorización de las tecnologías que nos ocupan será necesario la implementación de scripts y programas en el lado del cliente, que se ejecuten remotamente y devuelvan mediante un canal seguro el estado del evento.

---

[3] Simple Network Management Protocol: protocoo de monitorización básico de dispositivos de red.

La aplicación NRPE[4] va a permitir esto último. Gracias a NRPE vamos a ser capaces de ejecutar cualquier programa remoto tanto en máquinas Linux como Windows, lo que va a permitir desarrollar código capaz de monitorizar todos los servicios que actualmente se ofrecen al cliente.

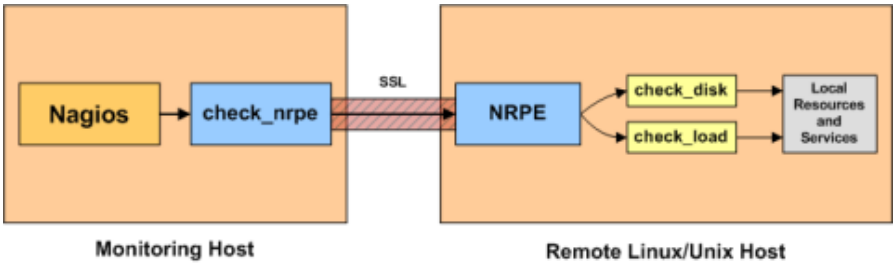

Ilustración 2 – Funcionamiento teórico de NRPE

Nagios también incorpora estadísticas con el fin de crear reportes del estado del sistema. Esto se realiza mediante el histórico de eventos, en base a los criterios internos de la aplicación.

De este modo es posible disponer de reportes relacionados, por ejemplo, con la disponibilidad de los *hosts* o la disponibilidad de los servicios de un determinado *host* o un determinado *servicegroup*.

El correcto uso de estos reportes y la planificación de los mismos pueden ayudar a identificar necesidades en cuanto a la capacidad y disponibilidad de los errores en función de las gráficas y resultados obtenidos.

Por último Nagios ofrece la posibilidad de planificar el periodo de *downtime*[5] de un *host* o *service*, con el fin de que una parada programada no influya negativamente en los resultados de estos reportes.

Nagios incorpora una pequeña interfaz gráfica para la visualización de los datos monitorizados. Se ha optado para este estudio utilizar un proyecto basado en Nagios que ofrece una interfaz Web más elaborada y atractiva, y que además incorpora a las funcionalidades de Nagios otras que nos ayudarán a la Gestión de la Disponibilidad y Capacidad de los servicios.

Ninja, además de esto, nos va a permitir transformar la implementación de Nagios. En Nagios las configuraciones y resultados se almacenan por defecto en ficheros de texto plano. Ninja dispone de un módulo conocido como Merlín que almacena esta información en bases de datos y tablas de mysql, de manera que se mejora el almacenaje, la seguridad y la gestión de los datos.

La interfaz presenta ventajas respecto a la presentación de la información, siendo de gran utilidad las distintas visualizaciones de grupos que se ofrecen y que mejoran la percepción del operador en infraestructuras complejas:

---

[4] Nagios Remote Plugin Executor: ejecución remota de scripts bajo demanda.

[5] Periodo de tiempo en el que un determinado servicio u *host* no es operativo, ya sea no intencionado o programado.

- **Tactical Overview**: muestra un resumen general del estado de servicios, *host*... aporta una visión directa y genérica del estado de la infraestructura.
- **Host y Service Detail**: presenta todos los objetos sin filtrado ni agrupamiento previo.
- **Hostgroup y Servicegroup Summary**: muestra el estado general de los miembros del grupo, sin mostrar cada miembro.
- **Hostgroup y Servicegroup Overview**: muestra los distintos grupos existentes, y los miembros del grupo con el estado de los servicios de cada miembro, sin mostrar el nombre de los servicios.
- **Hostgroup y Servicegroup Grid**: muestra los distintos grupos existentes, y los miembros del grupo con el estado de los servicios de cada miembro, mostrando el nombre de cada servicio relacionado.

Existen además otras vistas e ítems relacionados con los distintos objetos de Nagios.

Por otro lado, Ninja ofrece varios tipos de *reporting* basados en los cálculos internos de Nagios y en otros cálculos que añaden el módulo Merlin:

- **Trends**: gráficas lineales en base a los objetos de Nagios; *hosts*, *hostgroups*, *services*...
- **Alert history**: muestra todas las alertas que responden a los criterios definidos.
- **Alert summary**: resumen de alertas englobadas en unos criterios.
- **Notifications**: notificaciones mandadas mediante los mecanismos establecidos (frecuentemente correo electrónico).
- **Availability**: reporting en base a la disponibilidad de los objetos. Útil para el reporting de Gestión de la Disponibilidad.
- **SLA Reporting**: reporting en base a los SLA's[6] pactados con el cliente. Útil para la Gestión de Niveles de Servicio.

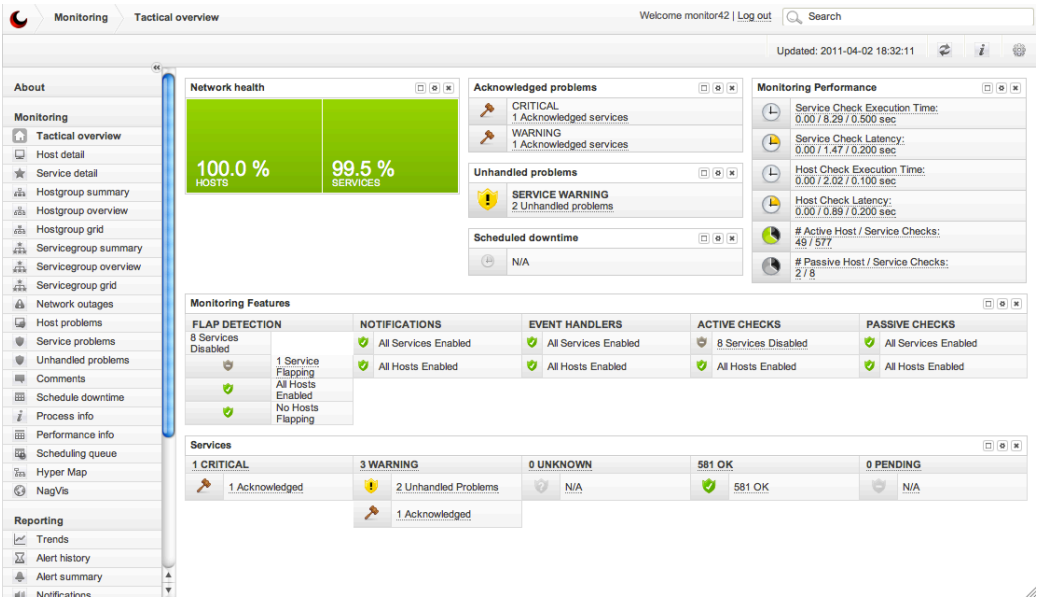

Ilustración 3 – Ventana principal de Ninja

6 Service Level Agreement: acuedo del nivel de calidad del servicio que debe ofrecer la organización IT.

Gestión de la Capacidad y la Disponibilidad en ITIL v3

Information Technology Infrastructure Library (ITIL) es el conjunto de prácticas y definiciones escogidas para gestionar todos los procesos relacionados con el área de las tecnologías de la información. Con la incorporación de ITIL se intenta *procedimentar* y gestionar cualquier proceso actualmente en funcionamiento o que deba ponerse en marcha para la buena gestión del servicio.

Teóricamente, ITIL se concibe como un ciclo de vida circular del servicio, que se ve retroalimentado por los resultados obtenidos de cada paso anterior. Las etapas de este ciclo son:

- **Estrategia del servicio**: estudio de mercado y de la oferta de servicios innovadores y/o requeridos por el cliente.
- **Diseño del servicio**: análisis de viabilidad del servicio y detalle teórico del mismo.
- **Transformación del servicio**: proceso de reconversión del servicio en base al diseño establecido.
- **Operación del servicio**: devolución del servicio totalmente operativo, definido y funcional.

Una quinta tarea está presente durante el ciclo circular del servicio, y tiene retroalimentación directa y constante con las demás tareas:

- **Mejora continua del servicio**: estudio de cambios y necesidades del servicio para la tarea actual o futura.

Ilustración 4 – Ciclo de Vida ITIL v3[7]

[7] Ilustración de http://www.educationfive.com/

Los distintos procesos de ITIL se enmarcan dentro de cada una de las etapas del servicio. Cada proceso se encarga de la gestión de un área distinta de la empresa TI, y su cometido está vinculado con la etapa en la que se encuentra cada uno de ellos.

Los procesos relacionados con el sistema de monitorización son los de Gestión de la Disponibilidad y Gestión de la Capacidad del servicio. Estos procesos se encuadran dentro de la etapa de Diseño del Servicio.

- **Gestión de la Disponibilidad y Continuidad del Servicio**: el cometido principal de este proceso es asegurar que se cumplen los niveles de disponibilidad y continuidad reflejados en los *Service Level Agreement* (SLA) pactados entre el cliente y la empresa TI.

  Con el fin de asegurar la continuidad del servicio para el usuario final, es necesario realizar una serie de definiciones del proceso; conocer las métricas para el control de la disponibilidad, comprobar la disponibilidad; con sistemas de monitorización que aporten datos veraces, definir planes de prueba para sistemas de alta disponibilidad y planes de *disaster and recovery*[8], entre otras tareas.

- **Gestión de la Capacidad**: proceso encargado de estudiar y satisfacer la capacidad que necesita el cliente, a todos los niveles; capacidad de almacenamiento, capacidad de procesamiento…

  Uno de los aspectos más importantes en el proceso, es ofrecer una capacidad correctamente dimensionada con las necesidades del cliente. Las tecnologías implementadas deben satisfacer las necesidades de los usuarios y de la previsión de crecimiento, sin llegar a ser inferior (en cuyo caso se pueden dar problemas de disponibilidad y continuidad) y sin ser excesivamente superior (suponiendo un gasto innecesario para el cliente).

El proceso de Gestión de la Disponibilidad y Continuidad del Servicio depende claramente del sistema de monitorización. El sistema actual, y el proyectado, deben ser herramientas para supervisar la disponibilidad del servicio. Estos datos serán utilizados tanto por la misma empresa TI; que debe estudiar proactivamente las carencias del servicio y reactivamente solucionar lo más rápido posible las incidencias, y también al cliente; que requiere una herramienta capaz de ofrecer datos calculados de la calidad del servicio.

Respecto a la Gestión de la Capacidad, las herramientas de monitorización deben suponer un arma proactiva para el estudio de tendencias de uso y de dimensionamiento del servicio. Gracias a los gráficos y a la supervisión de incidencias podemos descubrir problemas de capacidad que requieran una inversión para subsanarlos, o simplemente un rediseño parcial del servicio.

Existe otro proceso que, pese a no tener una relación directa con el sistema de monitorización, es necesario introducir, ya que es uno de los procesos base en los que se apoya toda la gestión del servicio: la Gestión de Niveles de Servicio.

---

[8] Protocolos de actuación ante un desastre que afecte a los servicios y cómo deben restaurarse.

- **Gestión de Niveles de Servicio**: el proceso de Gestión de Niveles de Servicio es el encargado de definir una serie de objetivos en la calidad del servicio. Se definen para ello tiempos de respuesta, de resolución de incidencias… a la vez que porcentajes de disponibilidad.

  Los niveles de servicio (SLA), deben ser cumplidos por parte de la empresa TI a la hora de prestar sus servicios. Por ejemplo, se deben pactar tiempos de respuesta y resolución ante un problema de disponibilidad de Exchange. Deben estar definidos tanto el *downtime* asumible a lo largo de un año y los umbrales de satisfacción que contempla el cliente.

Conclusiones

En lo referente a MOM, las interfaces *Operator Console* y *Administrator Console* engloban de forma clara las gestiones que podemos realizar con MOM.

La complejidad de la solución reside más en la complejidad de la infraestructura que en la solución misma. Es por ello que en diseños con una infraestructura extensa, y/o en diseños con una dispersión alta de servicios a monitorizar, la dificultad de definir y gestionar aumenta de forma exponencial.

Es necesario por ello realizar un estudio muy minucioso del escenario de implementación, ya que de nada sirve poseer de sistemas de monitorización potentes que lleguen a ser imposibles de administrar por la cantidad de alertas y eventos que se reciben continuamente.

Por ejemplo, en un cliente grande, con un parque extendido y amplio pueden darse condiciones o dificultades que provoquen eventos erróneos en el sistema de monitorización. Muchas veces es necesario discriminar eventos que, pese a tener una relevancia baja o media, son situaciones cotidianas con las que deben convivir los administradores y que estorban más que ayudan en la supervisión del estado de la infraestructura.

La solución de Nagios + Ninja resulta atractiva desde un punto de vista teórico. Si se dedica el suficiente tiempo para el estudio de la migración y para el pulido de alertas y eventos en el nuevo sistema, este puede dar resultados coherentes, que pese a no llegar al nivel de detalle de MOM, son válidos para la empresa TI y en ilación, también para el cliente.

La estructura de objetos de Nagios puede utilizarse para interpretar e implementar los distintos mecanismos que ofrece MOM. Esto ayudará a disponer de un sistema similar al anterior una vez completada la migración.

Respecto a la implementación interna, disponer de un módulo que añada la característica de almacenar la información en una base de datos, acerca también este aspecto a la implementación actual de MOM. Tal vez este punto parezca trivial, pero posteriormente cuando sea necesario tratar la información mediante consultas SQL o programar tareas de *backup* remotas será útil disponer de un Sistema de Gestión de Bases de Datos (SGBD).

Con la entrada de ITIL se añade una pregunta, ¿Es útil para una empresa TI implementar modelos de gestión o simplemente es una herramienta de control y supervisión para el cliente? ITIL propone un modelo que puede ser complejo de implementar, pero que puede aportar claridad a la gestión del servicio.

No olvidemos que la finalidad primordial de un sistema de monitorización debe ser detectar problemas en cualquier servicio de la infraestructura, para poder reaccionar de forma ordenada o incluso anticiparse a los problemas.

En esto ITIL puede ser un gran aliado, ya que "obliga" a la empresa TI a estudiar y contemplar mecanismos de *proactividad* para impedir fallos críticos o catastróficos, por ejemplo; un plan de disponibilidad puede probar que un clúster no se comporta como debe, o descubrir la necesidad de disponer de procedimientos de contingencia y *rollback,* para impedir que el desconcierto y el nerviosismo se apoderen de situaciones críticas.

Análisis del sistema actual y objetivo

Introducción

Se va a abordar en este punto el estudio de los sistemas actuales y objetivo, con el fin de conocer la implementación concreta y particular que tienen y deberán tener ambos sistemas. Este estudio servirá como base para disponer de un protocolo de migración seguro y orientado a las necesidades de la infraestructura.

Ante el proyecto de migración a Nagios, surgen varias preguntas que debe plantearse necesariamente la empresa IT:

- ¿Es posible migrar MOM a Nagios?
- ¿Es posible replicar el comportamiento de MOM en Nagios?
- ¿Es posible reaprovechar la información de MOM en Nagios?

Estas y otras preguntas requieren un estudio serio y profundo de las dos herramientas, pero sobretodo de la herramienta MOM, ya que al tratarse de un software de código cerrado resulta desconcertante saber si va a ser posible, o no, disponer de la información necesaria para la migración.

En este punto del proyecto vamos a abordar tres fases de estudio de la herramienta MOM:

- **Estudio del comportamiento de la interfaz gráfica:** si queremos que el sistema objetivo sea similar en comportamiento al anterior, debemos conocer cómo se ofrece la información al operador.
- **Estudio de las bases de datos:** la información de MOM se almacena en SQL Server, por lo que vamos a poder realizar consultas para disponer de estadísticas, métricas y de información que mediante la interfaz no es posible extraer.
- **Estudio de los *Management Packs*:** Microsoft ofrece las definiciones en Management Packs, bajo la extensión *.akm. Estos archivos son documentos compilados y pueden extraerse a *.xml con una herramienta propia de Microsoft. El estudio de estos *.xml nos va a permitir conocer mejor cómo funciona MOM internamente y qué información emplea.

Además de la redacción de esta parte software, va a ser necesario conocer la distribución de la infraestructura, tanto de los servidores existentes como de los roles de cada uno de ellos.

Tras este estudio ya estaremos en condiciones de afrontar el estudio del sistema objetivo; Nagios. Gracias a la amplia documentación de Nagios propia de una herramienta de software libre, nuestros esfuerzos van a estar dirigidos a la transformación de los datos obtenidos en MOM a Nagios.

Esta transformación no va a ser una “copia” del funcionamiento de MOM (ya que como hemos visto en puntos anteriores, replicar totalmente MOM sería un fracaso) sino que debe considerarse como la creación de un sistema

totalmente nuevo, con unas bases de conocimiento de la infraestructura y del sistema anterior bien documentadas.
Para ello, vamos a proponer varias fases de estudio:

- **Transformación de objetos:** transformación de las entidades de MOM a objetos Nagios.
- **Transformación de eventos:** transformación de los mecanismos de control de MOM a consultas Nagios.
- **Transformación de operación:** calibrado del envío de alertas, registro de notificaciones… orientado al operador del sistema objetivo.

Una vez afrontadas estas fases, debe existir un periodo de convivencia entre ambos sistemas, y un estudio del comportamiento del sistema objetivo. Es posible que al transformar el sistema se hayan replicado eventos inútiles para el operador, o eventos que dentro del funcionamiento de MOM se ocultan o agrupan, y en el nuevo sistema saturan y dificultan la operación.

En este caso, el equipo dedicado a la migración debe estudiar el por qué de este comportamiento, y los cambios que deben realizarse para solucionarlo, teniendo en cuenta no empeorar el funcionamiento de la nueva herramienta.

Por último, debemos tener en cuenta la gestión del servicio en el sistema objetivo. La implantación de los procesos de la Gestión de la Disponibilidad y de la Gestión de la Capacidad no requieren migración, puesto que en la herramienta MOM no se contemplaba.

La adecuación de la herramienta Nagios a la gestión del servicio pasa por configurar reportes para la monitorización de estos dos procesos. La monitorización se va a realizar respecto al estado de algunos eventos y de algunos equipos.

Será necesario modificar los resultados obtenidos por estos eventos para disponer de datos fiables a entregar al cliente. Por ejemplo; es posible que un evento que monitoriza el estado de Active Directory informe cómo crítico cuando aparece cierto evento, y este estado crítico influya negativamente en el porcentaje de disponibilidad. Si el evento en realidad no influye en la disponibilidad del servicio, los datos finales a presentar ante el cliente no serán veraces, y además contraproducentes para la empresa TI.

Estudio de la infraestructura

Con el fin de tener una idea general de la infraestructura a monitorizar, vamos a realizar una presentación de los elementos hardware y software que comprenden la infraestructura del cliente.

El estudio de este proyecto engloba a los servicios orientados a la gestión del puesto de trabajo. Es por ello que la infraestructura objeto de estudio comprende servidores tanto Windows como Linux, y a lo servicios que estos ofrecen. No entran en el estudio la infraestructura de red para la interconexión de estos elementos.

Software

El software hace referencia a los distintos roles que se ofrecen al cliente. Los roles o servicios están implementados bajo tecnologías, que son al fin y al cabo las entidades a monitorizar en el sistema objetivo.

De todos modos, es positivo hablar de servicios en lugar de tecnologías a la hora de realizar este estudio, puesto que las prácticas ITIL exigen ofrecer un Catálogo de Servicios [9]independiente a las tecnologías que los implementen.

Servicio del catálogo	Tecnología
Servicio Directorio	Active Directory 2003
Servicio Documentos	Distributed File System
Servicio Gestión del Parque	SCM
Servicio Correo Electrónico	Exchange 2003
Servicio Colaborativo	WSS 3.0
Servicio Seguridad	McAfee
Servicio Monitorización	Nagios

Tabla 3 - Servicios del catálogo - Tecnología

Hemos nombrado los distintos servicios que se van a monitorizar y las tecnologías que deben monitorizarse en la puesta en marcha del sistema.

Todas las tecnologías salvo SCM y el propio Nagios se implementan sobre sistemas Windows Server 2003. Respecto a SCM y Nagios, la parte servidor (la que se va a monitorizar) se implementa sobre Red Hat 5.5.

Hardware

Una vez conocidos los servicios que se ofrecen al cliente, vamos a ver cómo se distribuyen estos roles sobre los servidores físicos de la infraestructura.

Los roles están distribuidos y replicados con el fin de ofrecer alta disponibilidad en los servicios. En la mayoría de los servicios existe un nodo central y los nodos departamentales distribuidos físicamente:

- **Servicios con nodos troncales y departamentales:** Directorio, Documentos, Seguridad y Gestión del Parque.
- **Servicios sólo con nodo troncal:** Correo Electrónico, Colaborativo y Monitorización.

[9] Lista de servicios de la organización TI ofrece a su cliente, y en la que se basa cualquier gestión.

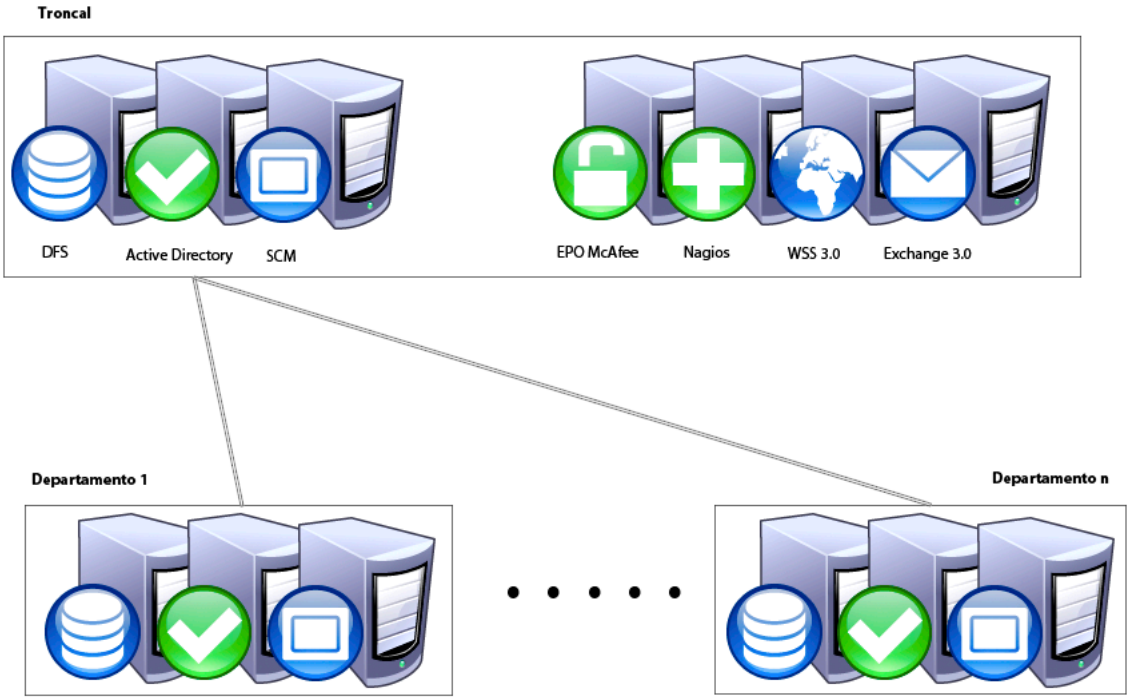

Ilustración 5 – Mapa de Infraestructura en el cliente

La información de los servicios que se ofrecen desde el nodo troncal únicamente no está replicada a lo largo de la infraestructura. En el caso de Exchange, el servicio sí que se implementa con un clúster a nivel de Exchange, pero desde el mismo lugar físico.

En cuanto a los servicios con diferentes nodos podemos encontrar dos funcionamientos básicos:

- **Replicación total:** el servicio de directorio implementado con Active Directory posee una réplica exacta en cada nodo, tanto troncal como departamental. De esto modo, si un servidor departamental no está disponible, el usuario se valida en el nodo troncal, y si este también está fuera de servicio, en el siguiente nodo departamental de menor coste.
- **Replicación parcial:** en el caso de los servicios de gestión del parque y gestión de documentos, los nodos departamentales sólo disponen de la información del departamento, y no del resto de la infraestructura. Es en el nodo troncal donde se guarda una copia de cada nodo departamental.

La infraestructura posee mayor complejidad que la reflejada aquí, pero esta visión general es suficiente para conocer el alcance de este proyecto.

Análisis del sistema actual de monitorización

Es necesario hacer un paralelismo realista entre MOM y Nagios: si MOM es un sistema base de monitorización, Nagios también lo es, pero no se encuentra ningún paralelismo para los Management Packs de MOM en Nagios.

Se hace evidente que ese conjunto de definiciones debe ser definido dentro de este proyecto, y que es necesario conocer un poco más sobre la herramienta MOM.

La pregunta a responder es sencilla; ¿Qué monitoriza MOM y cómo lo monitoriza?

Estudio de Event Rules mediante la interfaz gráfica

Intentamos responder a esta pregunta utilizando en primer lugar el entorno gráfico *Administrator Console*. La idea principal es conocer de qué se compone una *Event Rule*[10] de monitorización de MOM. Escogemos para ello, por ejemplo, varias *Event Rules* relacionadas con la monitorización de Active Directory para consultar sus características:

Event Rule *Replication has been stopped with a source*

La pestaña "General" de una *Event Rule* aporta información general de la regla. Entre los datos que encontramos en esta pestaña tenemos; el nombre de la regla, breve descripción, ubicación en la estructura de MOM, identificador único, y datos de última modificación.

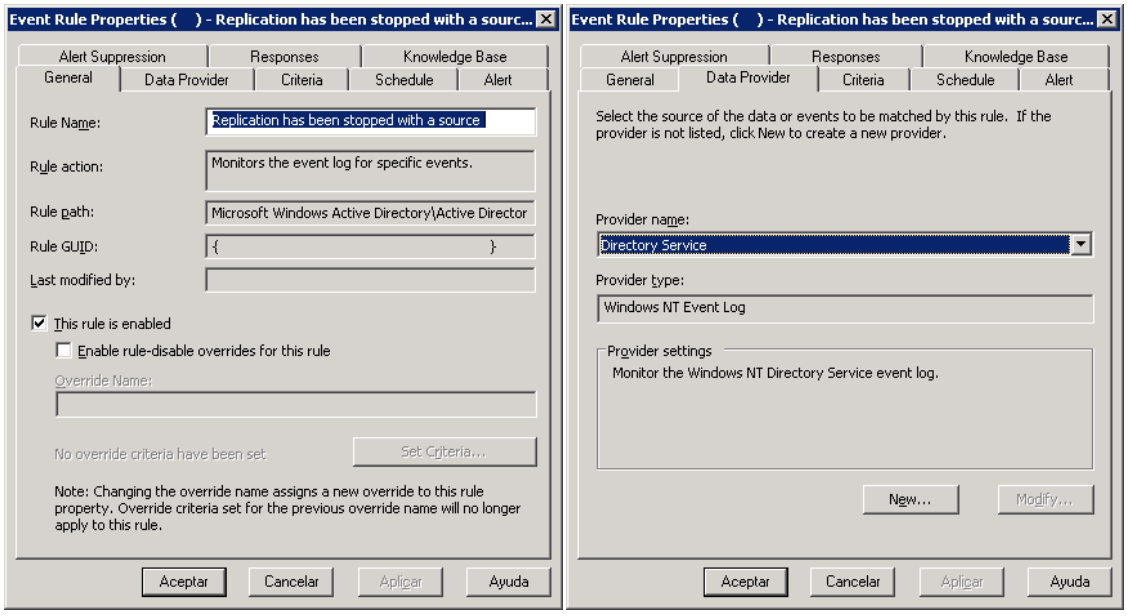

Ilustración 6 – Pestañas General y Data Provider

En la siguiente pestaña, "Data Provider", se aporta el primer dato interesante para el estudio; la fuente de consulta de la regla. En este caso de ejemplo nos percatamos que el tipo de fuente es "Windows NT Event Log", es decir, eventos del sistema Windows, particularmente del Event Log[11] "Directory Service".

---

10 Elemento básico de monitorización en MOM.

11 Registro de sucesos de Windows. Fuente de la utilidad "Visor de Sucesos" de Windows.

En “Criteria” se detallan las características de la consulta de la regla sobre la fuente definida en “Data Provider”. En este caso, al tratarse de una definición por Management Pack los criterios están predefinidos, pero podríamos crear o modificar una regla con todas las opciones que nos brinda Criteria:

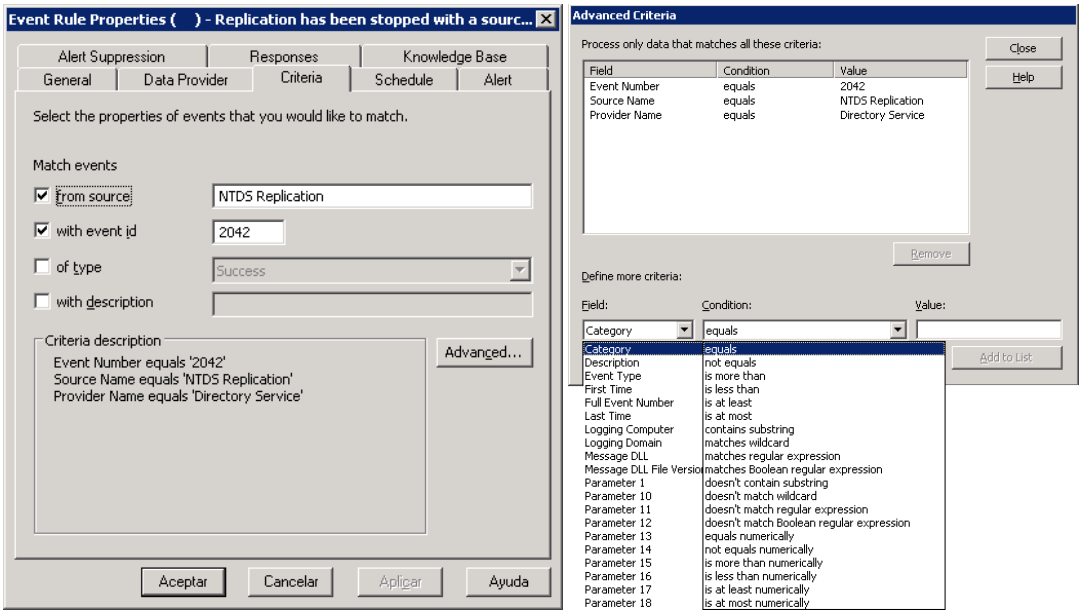

Ilustración 7 – Pestaña Criteria y subpestaña Advanced

La información que conocemos hasta este punto es que esta regla realiza una consulta sobre el Event Log relacionado con el directorio activo (Directory Service), en busca del evento con ID 2042 y *source* “NTDS Replication”.

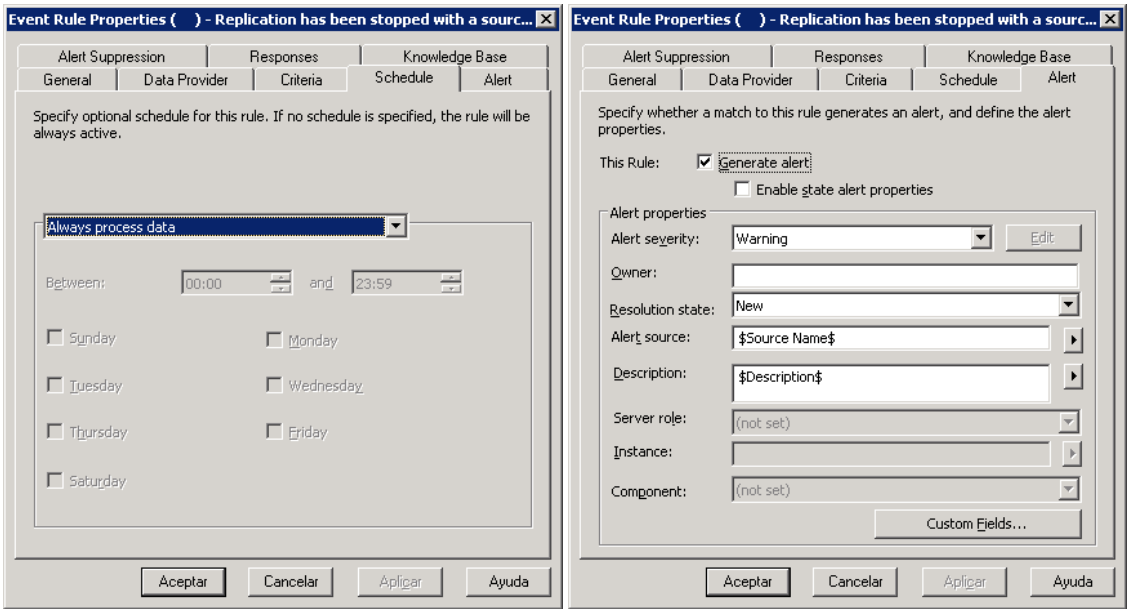

Ilustración 8 – Pestañas Schedule y Alert

Como podemos ver en este caso la pestaña “Schelude” tiene inactivas las opciones de configuración. Desde esta pestaña podemos configurar cada cuanto se realizará la consulta. Al tratarse de un evento interno de un sistema

Windows, esta regla está configurada como disparador, y se ejecutará cuando se detecte la aparición de un evento en el Event Log.

La pestaña “Alert” hace referencia a la alerta que se registrará para visualizarla en *Operator Console*. Este dato también será importante para el proyecto, ya que no todos los eventos tienen la misma severidad. En este caso, se registrará la alerta como un *Warning*, y mostrará los datos de los campos *Source* y *Description* del evento.

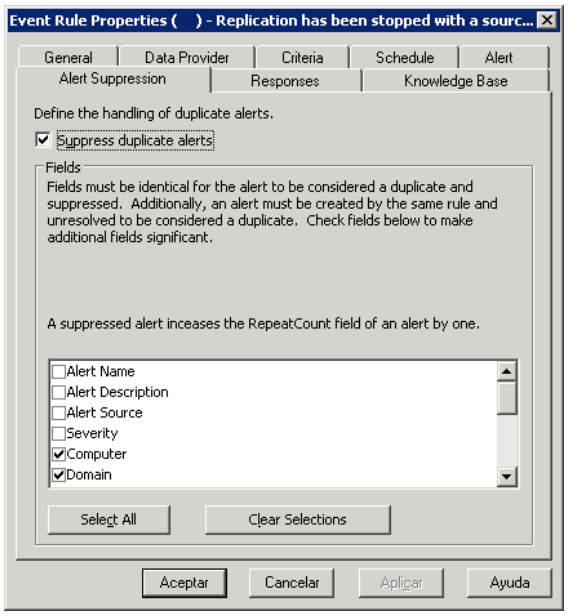

Ilustración 9 – Pestaña Alert Supressio

La última pestaña que nombraremos es la de “Alert Suppression” y también tiene importancia dentro del proyecto. Esta pestaña permite configurar cómo se sobrescribirán, se añadirán (o no) nuevas alertas si existe una anterior del mismo tipo. Esta opción es muy útil para reducir el número de alertas registradas a revisar por los operadores.

Event Rule *Script – AD Essential Services Running*

Tomamos otra regla de ejemplo para comprobar su funcionamiento. En este caso, la regla realiza una consulta mediante un script que puede o bien estar distribuido mediante los agentes de monitorización, o bien remotamente desde el mismo servidor.

Al explorar las pestañas “Data Provider” y “Criteria” podemos observar que no existen configuraciones para esta regla. Tendremos que abrir la pestaña de “Responses” para conocer un poco más:

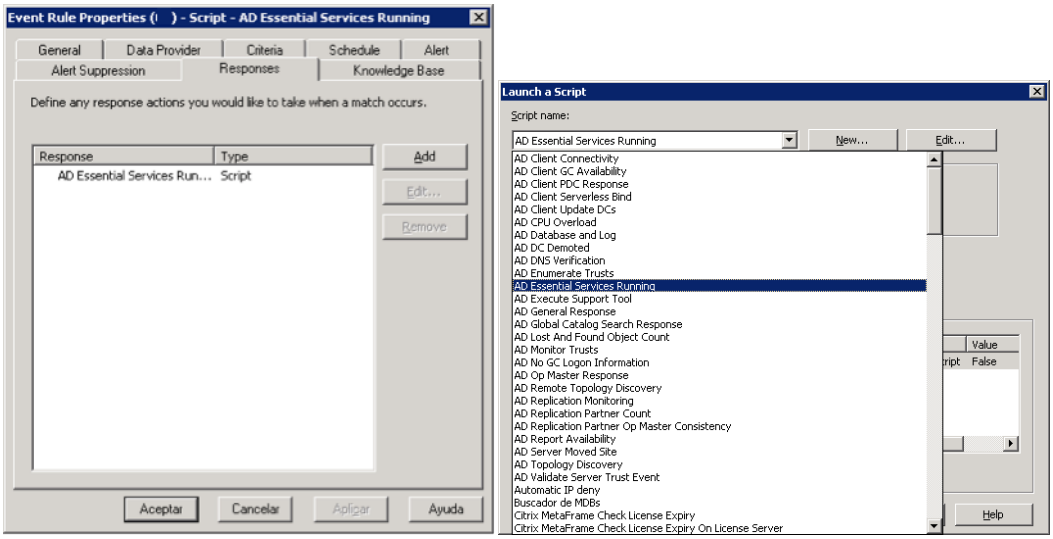

Ilustración 10 – Pestaña Responses y subpestaña Add

La regla lanza un script tipo Visual Basic Script (VBS) llamado "AD Essential Services Running" que consulta si los servicios necesarios para que funcione Active Directory están en funcionamiento.

Existen otros tipos de reglas a lo largo de los Management Packs instalados en MOM que consultan otro tipo de información de aplicaciones, logs, WMI[12].. pero la monitorización por Script y por Event Log suponen el 10% y el 75% respectivamente de las reglas activas en el cliente. Estos datos los conoceremos en la siguiente etapa del estudio.

Además de ser un número de reglas elevado, ambos tipos pueden ser relativamente sencillos de replicar en el sistema objetivo.

Estudio de Event Rules mediante SQL Server

Indexar todas las reglas que existen en MOM manualmente tal y como hemos hecho en el punto anterior es prácticamente imposible. La cantidad de reglas en el sistema que posee el cliente objeto de estudio asciende a 5000, contando todos los Management Packs instalados para la monitorización de todos los servicios del catálogo.

Es por lo tanto razonable plantear un mecanismo mucho más ágil que nos permita extraer estadísticas certeras sobre el funcionamiento de MOM e información útil para el sistema objetivo. Se pretende realizar esta tarea mediante consultas sobre SQL.

Tras revisar las numerosas tablas existentes en las bases de datos de MOM, encontramos las que realmente necesitamos para el dato que deseamos calcular:

- **dbo.ProcessRule:** tabla con todas las Event Rules instaladas.
- **dbo.ProcessRuleMembership:** relación regla – grupo de reglas.

---

12 Windows Management Instrumentation: estándar de acceso administrativo a través de la red.

- **dbo.ProviderInstance:** cada fila hace referencia a una fuente de información distinta (Event Log, Performance Monitoring, Generic Provider…)

Mediante la siguiente consulta SQL podemos extraer la relación regla – provider, de forma que la cuenta de todas las reglas con el mismo provider nos muestre la cantidad de reglas instaladas:

```
SELECT CRN.Name, C.Name
FROM dbo.ProcessRule C INNERJOIN dbo.ProcessRuleMembership CR ON
C.idProcessRule = CR.idProcessRuleMember INNERJOIN dbo.ProcessRule
CRN ON CRN.idProcessRule = CR.idProcessRuleGroup
WHERE C.idProviderInstance IN(SELECT idProviderInstance
FROM dbo.ProviderInstance
WHERE ClassID LIKE'IDENTIFIC_PROVIDER')AND CR.IsDeleted != 1 AND
C.IsEnabled = 1
ORDERBY CRN.Name
```

Donde `'IDENTIFIC_PROVIDER'` hace referencia al identificado único del Provider de MOM.

Tras realizar los cálculos oportunos, obtenemos los siguientes datos relacionados con las distintas fuentes de información de MOM:

Nombre	Nº reglas activas	Porcentaje
Event Log	3502	75,17%
Generic Provider	506	10,86%
Performance Monitoring	458	9,83%
Timed Event	157	3,37%
Application Log	28	0,6%
WMI Numeric Provider	5	0,11%
WMI Events	3	0,06%

Tabla 4 – Relación de reglas activas y porcentaje que suponen

La conclusión a la que podemos llegar con estos datos es que una gran parte del trabajo que realiza MOM consiste en recabar información ya existente en los Event Log del sistema.

La parte compleja y que aportan los Management Packs son las formas de tratar esta información, discriminando y catalogando los eventos de cada servicio según los criterios definidos por el fabricante.

Por otro lado, "Generic Provider" hace referencia a los eventos que extraen su información de scripts ejecutados en el agente o de forma remota. Muchas de estas reglas se pueden considerar como las reglas "esenciales" para la monitorización de servicios.

La fuente "Performance Monitoring", pese a tener un porcentaje considerable, no es útil para este proyecto, ya que hace referencia a las reglas de conteo de datos y están orientadas más hacia las estadísticas del estado del sistema que a la monitorización por alertas.

Las demás fuentes tienen un papel ínfimo en la monitorización, y frecuentemente se utilizan en Management Packs aislados.

Tras estas consultas sobre SQL se plantea un desafío: ¿Podríamos extraer más información de las reglas desde SQL? Sería bueno, por ejemplo, conocer qué eventos del sistema se consideran desde MOM como "Critical Error[13]", para monitorizarlos desde la herramienta Nagios. O bien qué realiza cierto script para intentar implementarlo como un evento de monitorización en el nuevo sistema.

Por desgracia, esta información se almacena en SQL en un tipo de dato IMAGE, un formato binario que no permite extraer la información que precisamente necesitamos.

Estudio de Management Packs de Microsoft

Tal y como se ha dicho anteriormente, es posible considerar a MOM como una estructura base de monitorización, que permite añadir las definiciones necesarias según las exigencias de nuestra infraestructura mediante Management Packs.

Esto significa que las definiciones de reglas, alertas, grupos… realmente están almacenadas en los Management Packs, los cuales se añaden o se instalan sobre MOM tras descargarlos del proveedor (en este caso Microsoft) bajo la extensión *.akm.

Microsoft proporciona en su paquete de herramientas "MOM Resource Kit" la utilidad "MP2XML.exe", que permite convertir los archivos cerrados *.akm a *.xml. De esta forma, es posible consultar, modificar o añadir información al Management Pack.

Esta utilidad nos va a permitir conocer un poco más acerca de las reglas instaladas actualmente en el cliente. Tomemos de nuevo como ejemplo el Management Pack de Active Directory que contiene las definiciones de monitorización de los controladores de dominio en el fichero "MicrosoftWindowsActiveDirectory.akm".

El fichero *.xml resultante tras ejecutar "MP2XML.exe" tiene un total de 36238 líneas. Este XML no dispone de un esquema XML que defina la distribución de sus datos. Optamos por utilizar Microsoft Office Excel, ya que permite abrir fichero xml e interpretar sus datos incluso cuando no existe un esquema.

La tabla resultante tiene 4507 filas, de las que realizaremos el análisis de información:

- Las primeras 1309 filas corresponden a definiciones propias del proceso de instalación del *.akm. En ellas se especifican las comprobaciones necesarias para añadir el paquete, los grupos de equipos que se deben crear y monitorizar, las distintas carpetas de reglas que se deben construir en la raíz.

---

[13] Grado de severidad máximo que es posible registrar en MOM.

- Todas las demás filas son las definiciones de reglas, correspondiendo muchas veces varias filas a una misma regla. Nos centraremos en estas filas.
- Existe información en el *.xml que no se representa en la tabla Excel (a partir de la línea 21699), y que corresponde a los scripts VBS que utiliza MOM para monitorizar las reglas basadas en scripts. Esta información también será objeto de estudio.

Utilizaremos las opciones de filtrado que brinda Excel para extraer las filas que necesitamos para este estudio.

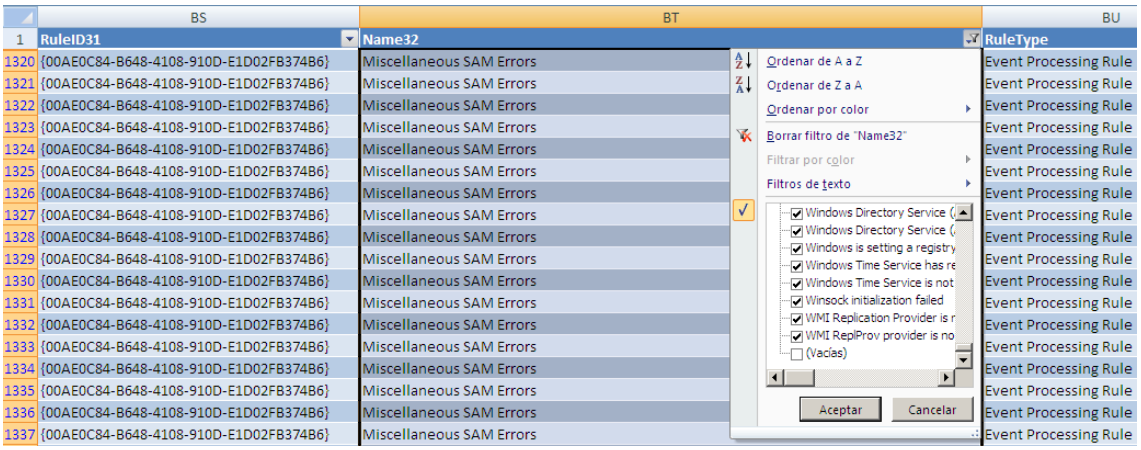

Ilustración 11 – Opciones de filtro de Excel

Estas son las columnas que filtraremos y estudiaremos:

- **Name32:** Nombre de la Event Rule que se muestra en MOM. Eliminamos todas las entradas que no contengan información.
- **RuleType:** tipo de regla, similar a la información del Provider. Supuestamente "Event Processing Rule" hace referencia a las reglas que se monitorizan por eventos.
- **ProviderID:** fuente de la información. En nuestro caso sabemos que el Provider con ID {907D456E-146C-11D3-AB21-00A0C98620CE} hace referencia al Provider Event Log.
- **Property:** referencia al argumento del filtro definido en Criteria, donde el valor numérico hace referencia a:
  - 01 – Event Number (ID)
  - 03 – Event Type
  - 04 – Message DLL
  - 05 – Source Name
  - 06 – Provider Name
  - 07 – Description
  - 13 – Username
  - 17 – Repeat Count

  En el caso de que un evento tenga varios argumentos en Criteria, existirá una línea en el Excel por cada argumento existente.
- **Value34:** valor del argumento referenciado en la columna Property. Por ejemplo; Property = 05, Value34 = NTDS General.
- **CompareType36:** condición que se debe cumplir entre el valor marcado en Value32, y el evento detectado:
  - 0 – Equals

    - 1 – Not equals
    - 2 – Is more than
    - 3 – Is less than
    - 4 – Is at least
    - 5 – Is at most
    - 6 – Contains wildcard
    - 7 – Matches wildcard
    - 8 – Matches regular expresion
    - 9 – Matches regular boolean expresion
    - 10 – Doesn't contains substring
    - 11 – Doesn't match wildcard
    - 12 – Doesn't match regular expresion
    - 14 – Doesn't match regular boolean expresion
    - 15 – Equals numerically
    - 16 – Not equals numerically
    - 17 – Is more than numerically
    - 18 – Is less than numerically
    - 19 – Is at least numerically
    - 20 – Is at most numerically
- **Pattern:** cuando el valor es una expresión booleana a tratar, en lugar de aparecer en Value34 aparece en Pattern.
- **GenerateAlert:** marca si se debe generar alerta o no (1 si, 0 no)
- **Severity:** severidad con la que se registra la alerta en el sistema de monitorización:
    - 20 – Information
    - 30 – Warning
    - 40 – Error
    - 50 – Critical Error
    - 70 – Service Unavailable

Con esta información podemos obtener datos muy interesantes para la monitorización en el sistema objetivo. Por ejemplo, podremos monitorizar dos eventos considerados por el Management Pack como "Critical Error", de los que conocemos sus ID's (404 y 1209), y unos 125 eventos con nivel de "Error" dentro de las definiciones del paquete, entre otras cosas.

Respecto a los scripts en VBS que hemos mencionado antes, también podemos recabar un poco de información sobre ellos. Estos scripts son complejos, y trabajan con variables globales internas de MOM, por lo que resultan complicados de entender a simple vista.

Sin embargo, Microsoft tiene documentado en KB's las descripciones de los scripts y una explicación general de qué realiza cada uno de ellos. Esta documentación nos ayudará a descartar los scripts que consideremos poco útiles para la monitorización, y a implementar scripts y programas que intenten reproducir el comportamiento de los scripts existentes en los Management Packs.

Por ejemplo, el script "AD Essential Services" según el KB de Microsoft monitoriza el estado de los servicios imprescindibles para Active Directory

(FRS, IsmServ, KDC, NetLogon, W32Time), además de comprobar si la ruta de SYSVOL[14] se ofrece como recurso compartido y es alcanzable.

Esta funcionalidad es sencilla de replicar a un script trivial que pueda utilizarse en el sistema objetivo.

Análisis del sistema objetivo

Vamos a estudiar el sistema objetivo que se basará en Nagios y la Interfaz gráfica de Ninja. El propósito de este punto es realizar un estudio más profundo de la solución de código abierto Nagios, para poder proponer en la siguiente sección un proceso de instalación en la migración.

La instalación y configuración del software necesario se documentará en el estudio de migración. Es necesario antes de ello saber cómo vamos a *parametrizar* el sistema en varios flancos:

- ***Parametrización* del servidor:** es necesario conocer la filosofía que va a tomar el proyecto para definir todos los objetos Nagios; estructuración de *hosts* y *hostgroups*, jerarquía, uso de plantillas, programación de *services*, periodos de alertas…
- ***Parametrización* de los agentes:** scripts y programas encargados de la monitorización desde el lado del cliente; monitorización por eventos, replicación de scripts en los servidores…

Traducción de elementos al sistema objetivo

Tras estudiar las entidades existentes en MOM, es necesario conocer los objetos equivalentes de cada uno de ellos en el sistema objetivo. Se propone la siguiente homologación:

MOM	Nagios
Managed Computers	Hosts
Computer Groups	Hostgroups
Event Rules	Services
Rule Groups	Servicegroups
Notifications	Contacts
Notification Groups	Contactgroups
Providers	Commands

Tabla 5 – Equivalencia de elementos MOM y objetos Nagios

Esta traducción es importante como punto de partida para el sistema objetivo; se va a utilizar una estructuración similar a MOM. Esta base es uno de los puntos que ayudarán al operador a manejarse sobre la interfaz de Ninja de una forma ágil y productiva.

El modelo de jerarquía de Nagios nos va a permitir estructurar en distintos *hostgroups* y *servicesgroups* la infraestructura. Por ejemplo, definiremos

[14] Volumen del sistema es un directorio compartido, que contiene información esencial del dominio, como por ejemplo las políticas de grupo.

grupos de *hosts* enfocados a los servicios (DFSSERVERS) pero también grupos de *hosts* orientados a la distribución geográfica de la infraestructura (DEP01, DEP02 …).

Un servidor por tanto puede encontrarse al final de la jerarquía de *hostgroups* anidados, pero a su vez puede formar parte de otra jerarquía. El operador tiene la posibilidad de encontrar la información de este servidor según lo que realmente está buscando.

En cuanto a las plantillas, vamos a utilizarlas para incorporar configuraciones comunes en distintos objetos. Por ejemplo, se considera útil que por defecto todos los servidores incorporen el grupo de contactos "Operadores CSR", encargado de la monitorización 24x7.

Frecuencia de monitorización

Como vimos en el estudio de MOM, muchas monitorizaciones se realizan por disparadores en lugar de cada cierto intervalo de tiempo. La migración de este comportamiento a Nagios no es para nada trivial, y resulta más sencillo definir un periodo fijo que afecte a algunos *services* (por ejemplo, la monitorización de los Event Logs).

Por otra parte, Microsoft dispone de documentación de cada cuanto se ejecutan determinados scripts de los Management Packs. La programación en Nagios no tiene porque ser un calco a estas definiciones, pero es un buen punto de partida para saber cómo de necesario es cada script.

Script	Rule	Default Frequency
AD CPU Overload	Script – AD CPU Overload	Once per minute
AD Database and Log File	Script – AD Database and Log File	Every 15 minutes
AD DNS Verification	Script – AD DNS Verification	Once per day
AD Essential Services	Script – AD Essential Services Running	Every 11 minutes
AD General Responses	Script – AD General Responses	Every five minutes

Tabla 6 – Ejemplos de frecuencia de ejecución en el MP de Active Directory

Esta cuestión es muy importante. Nagios internamente sabe repartir sus tareas a lo largo del tiempo; sabe programar los servicios para que no se solapen entre ellos y no se ejecuten varios sobre el mismo servidor al mismo tiempo.

Pero pese a ello, muchos servicios pueden ocupar más tiempo del esperado en responder, solaparse con otros servicios y consumir más recursos de los deseados en un servidor. El sistema de monitorización puede pasar de ser el encargado de detectar fallos críticos, a ser el causante de los mismos.

Es necesario establecer medidas para evitar estas situaciones:

- Realizar laboratorios, preproducción y pilotaje de las soluciones.
- Scripts y programas C# muy depurados.

- Scripts y programas C# livianos.
- Monitorización de elementos importantes; despreciar servicios que no aporten información útil.

Sistema de monitorización por agente

Se pretende utilizar la monitorización desde el lado servidor hasta lo que resulte alcanzable (monitorización de hardware, discos, memoria, servicios del sistema operativo…). La monitorización desde es servidor aporta ventajas de rendimiento sobre la monitorización por agente.

Sin embargo, será necesario implementar soluciones que trabajen en el lado cliente para monitorizar las tecnologías implementadas en la infraestructura.

Se ha optado por el lenguaje de programación C# que dispone de librerías suficientes para la inspección de los elementos a monitorizar. Estos programas tienen como principal objetivo no suponer una carga excesiva para el sistema, ya que de nada sirve detectar un servidor comprometido comprometiéndolo más todavía.

El programa que intentará monitorizar cerca del 75% de lo que realiza MOM, inspeccionará los Event Log según el servicio y partiendo de la información extraída de los Management Packs en el punto anterior.

```
string Query = "SELECT * FROM Win32_NTLogEvent WHERE Logfile = 'Directory
Service'";
foreach(string id in ids)
{
      Query += " OR Eventcode = " + id;
}
```

El programa realiza consultas WMI sobre el sistema operativo, y busca los EventID's definidos por el Management Pack. Si encuentra alguno de ellos, el programa devuelve el error y lo registra en Nagios con la severidad pertinente (también definida por el Management Pack).

Otros scripts apoyarán el trabajo de este programa mediante la replicación de los scripts en VBS que como dijimos, también se encuentran dentro del propio archivo del Management Pack.

En el caso de Active Directory, por ejemplo, conocemos la importancia de comprobar el funcionamiento de todos los servicios del sistema relacionados con AD, que el recurso compartido de SYSVOL sea accesible, que la replicación entre los distintos controladores se efectúe dentro de los tiempos establecidos, que el proceso de AD LSASS.EXE[15] no exceda el consumo de CPU establecido…

El estudio de los scripts incluidos en un Management Pack, y la replicación de aquellos que debemos seleccionar como esenciales, ocupará una gran parte en el tiempo de migración de MOM a Nagios. Es necesario, por lo tanto, definir un calendario o cronograma que permita establecer los scripts más prioritarios al

[15] Proceso que representa el consumo de memoria y CPU de Active Directory en el sistema.

principio, y dejar los secundarios para añadirlos más tarde al sistema de monitorización.

Distribución y replicación de software a los agentes

Una de las tareas que MOM realiza de forma interna y transparente para el operador es la distribución y replicación de la información necesaria para la monitorización desde el lado del agente.

Cuando un determinado servidor pertenece a un grupo de reglas, MOM distribuye a ese servidor (al agente que tiene instalado) la información necesaria para monitorizar dichas reglas; definiciones generales, disparadores de los Event Logs, scripts…

En Nagios no existe un sistema de serie que permita olvidarse de la tarea de distribución y actualización del sistema de monitorización por agente. El operador debe replicar manualmente toda la información necesaria (en este caso, scripts para NRPE y archivos de configuración), lo que resulta cuanto menos tedioso en una infraestructura relativamente compleja.

El funcionamiento ideal pasaría por realizar modificaciones en Nagios para que según la relación objetos – servicios, distribuyera automáticamente los scripts y la configuración necesaria desde el propio servidor de monitorización hacia los sistemas monitorizados.

Esta solución escapa al alcance de este proyecto, puesto que cualquier modificación del sistema base Nagios prolongaría el proyecto demasiado tiempo. Es necesario plantear una alternativa menos compleja.

Al igual que se han programado las aplicaciones de monitorización en C#, se pretende programar una aplicación gráfica en C# que en base a los archivos de configuración de objetos de Nagios permita distribuir scripts y archivos de definiciones necesarios para la monitorización.

Por ejemplo, a todos los equipos pertenecientes al *hostgroup* DFSSERVERS, se les distribuye el programa check_event_log.exe, el archivo de definiciones DFSReplication.csv y en el archivo de configuración de NRPE se añade la definición del evento con el nombre, y la sentencia a ejecutar.

Análisis de resultados y supresión de alertas

Otro asunto a tratar es cómo Nagios debe interpretar los resultados de los distintos *services*, es decir; cómo una serie de eventos por separado deben ofrecer una visión única sobre la disponibilidad y el estado de los servicios del cliente.

El estado general de un determinado servicio debe ser realista. Para ello es necesario refinar muy a fondo el comportamiento de cada *service*, y además, incluir objetos del tipo *servicedependency*, que permiten definir dependencias entre varios servicios; si un determinado *service* no tiene un determinado estado, no se lanza otro *service*.

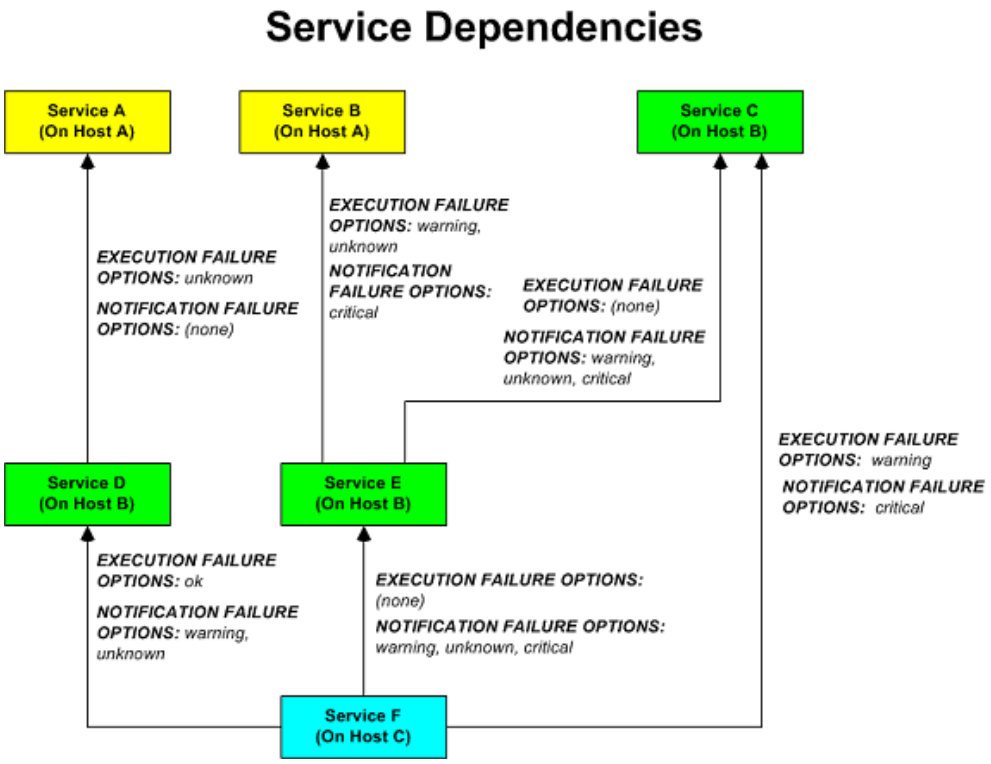

Ilustración 12 – Ejemplo de dependencias entre servicios

La interpretación de los resultados de *services* y las dependencias tienen un papel importantísimo en la adecuación del sistema de monitorización:

- **Gestión de la Disponibilidad y Capacidad:** tal y como hemos mencionado a lo largo de este documento, a los distintos procesos relacionados con la monitorización debemos proporcionarles resultados verídicos del estado de los servicios ofrecidos.
- **Supresión de alertas:** como dijimos en el estudio de MOM, la supresión de alertas inútiles o repetidas es un factor clave para no saturar el proceso de revisión de la herramienta.

Conclusiones

Concluimos esta sección donde hemos analizado la situación actual de la infraestructura, el sistema actual de monitorización y la propuesta de sistema objetivo.

Tras el estudio del sistema que actualmente se encarga de la monitorización, resulta más evidente la complejidad y sofisticación de una herramienta propietaria como es *Microsoft Operations Manager*. Esta herramienta permite diseccionar de forma muy eficaz la parte administrativa de la propia herramienta de la parte de operación.

Se trata en este estudio con infraestructuras complejas, que requieren una administración intensa de sus sistemas. Poseer de sistemas de monitorización que prácticamente funcionan *out-of-the-box* permiten a los operadores, y

administradores en general dedicarse a las tareas que realmente deberían requerir esfuerzo; la solución de las incidencias y problemas de los sistemas.

Los Management Packs son un elemento importantísimo dentro del sistema. Tal vez esta apreciación se hace más indiscutible cuando, una vez dentro del sistema objetivo, no encontramos una solución equivalente que nos permita dedicarnos a configurar el sistema de monitorización sin tener que estudiar técnicamente cada uno de los servicios ofrecidos al cliente.

Sólo el estudio de uno de los Management Packs, como es el de Active Directory, supone una cantidad de horas alta dentro del proyecto. Debemos comprender el funcionamiento técnico por una parte y tras esto, implementar soluciones que permitan replicar lo mejor posible este comportamiento.

Con el propósito de minimizar esta dedicación de horas se ha optado por investigar las distintas alternativas que existen en la comunidad para monitorizar los servicios implementados sobre el cliente. Podemos encontrar un repositorio extenso de scripts NRPE dedicados a la monitorización de estos servicios.

Pero la realidad es que desconocemos en muchos de estos scripts y programas, cómo están implementados, que grado de depuración tienen y cuanto pueden comprometer nuestros sistemas.

El agente NRPE desplegado sobre los servidores a monitorizar ejecuta cualquier script con credenciales SYSTEM[16], por lo que la organización debe analizar con mucho detalle qué permite ejecutar en sus sistemas.

El proceso de migración entre ambos sistemas por lo tanto, va a tener que diferenciar entre la instalación y configuración básica del sistema; donde se creará la estructura de Nagios y se monitorizará los elementos básicos (ping, espacio de disco, espacio de memoria…), la implementación y configuración de la monitorización de servicios; donde se programará los distintos scripts de monitorización NRPE, y la reconfiguración del sistema objetivo; un proceso que mediante las correcciones detectadas por los operadores debe definir el nuevo sistema dentro de los niveles exigidos.

16 Usuario del sistema con todos los privilegios sobre los recursos locales y de dominio.

Estudio de la migración del sistema

Introducción

Se han introducido los sistemas actuales y objetivos en un nivel teórico para la comprensión de este documento. Más tarde, se ha detallado el estudio de la herramienta actualmente implementada y cómo debería funcionar el sistema objetivo.

Es momento en este punto de definir el proceso de migración entre ambos sistemas en base a los conocimientos adquiridos en las secciones anteriores. Como se ha dicho, el proceso de migración constará de una serie de pasos para la instalación, configuración y definición de la herramienta.

Estos pasos en líneas generales son los siguientes:

- **Instalación y configuración básica del sistema:**
  - Instalación de Nagios y Ninja.
  - Exportar datos de MOM (servidores monitorizados, grupos a los que pertenecen).
  - Definición de los objetos básicos de Nagios (*host*, *hostgroups*, contactos y servicios base).
- **Implementación y configuración de monitorización de servicios:**
  - Programación de scripts: monitorización por Event Log y otros scripts.
  - Extracción de Event Rules para la monitorización por script.
  - Implementación de NRPE en sistema objetivo y distribución masiva a través de la infraestructura.
  - Revisión de funcionamiento y de reporte de alertas.
- **Reconfiguración del sistema objetivo:**
  - Configuración de dependencias y resultados globales para informes de disponibilidad y capacidad.
  - Monitorización conjunta de sistema actual y objetivo: estudio del comportamiento.
  - Reconfiguración en base al estudio continuo del sistema: corrección de estados erróneos, eliminar reglas innecesarias…

Estudio de la migración del sistema

La infraestructura de monitorización consta de un único servidor físico ubicado en el CPD central de la organización. Ya que no se monitorizan elementos de red, un único servidor de monitorización se considera dimensionado para los servicios actualmente implementados.

El servidor físico tiene la siguiente configuración hardware:

```
2 procesadores Intel Xeon E5520
8GB memoria DDR2 667
4 SAS internos de 146GB y 15krpm
2 puertos GBE intergrados
```

Nagios se implementará sobre Linux, concretamente sobre el sistema operativo Red Hat Enterprise Linux Server, sin entorno gráfico como medida de seguridad y por ofrecer sus servicios mediante una aplicación Web.

La política de seguridad del cliente para sistemas Linux es dejar habilitado tanto el Firewall como *SELinux*[17] en los mismos, y realizar modificaciones en el sistema según las exigencias del producto.

Por otra parte, la política de seguridad del cliente para cualquier sistema, ya sea Linux o Windows, es bloquear por Firewall de red cualquier comunicación con el servidor, ya sea entrante o saliente, y abrirlas en base a las necesidades que requiera el proyecto.

Es por ello que en un principio se solicitan los puertos necesarios para el funcionamiento básico del sistema (actualización e instalación de software, SSH, NTP…) y para el funcionamiento de Nagios (aplicación Web en el 80, SNMP en 161 y 162…).

Posteriormente será necesario solicitar además los puertos para la monitorización por NRPE (5666), y otros que aparezcan en el transcurso del proyecto (puerto 25 para envío de correos, por ejemplo).

Instalación de Nagios y Ninja

Se va a utilizar la documentación oficial de Nagios para realizar la instalación de la herramienta. Existen repositorios como el de *rpmforge* que incluye los paquetes necesarios para la instalación, pero por motivos de seguridad se ha optado por seguir la documentación oficial y compilar las fuentes de Nagios en el servidor.

Es necesario registrar previamente el servidor Red Hat para habilitar los repositorios oficiales. Esta acción ya se ha realizado con anterioridad con el comando *rhn_register*.

La siguiente instrucción instala las dependencias mínimas para poder compilar el paquete de Nagios:

```
yum install httpd php gcc glibc glibc-common gd gd-devel
```

Creamos el usuario y el grupo necesarios para el funcionamiento del sistema interno de Nagios:

```
useradd –m nagios
passwd nagios (definimos una contraseña segura)
groupadd nagcmd
usermod –a –G nagcmd nagios
usermod –a –G nagcmd apache
```

Descarga del código fuente de Nagios, y de sus *plugins*:

---

17 Capa de políticas de seguridad en sistemas Unix desarrollado por la NSA.

```
wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-3.2.3.tar.gz
wget http://prdownloads.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.11.tar.gz
```

Descomprimimos el archivo de Nagios y ejecutamos la instrucción de configuración con el grupo dedicado a Nagios:

```
./configure --with-command-group=nagcmd
```

Tras esto, recibimos por pantalla información importante relacionada con la configuración de Nagios, como son el usuario, el grupo, la ruta de instalación...

Tras configurar el código fuente, ya es posible realizar la compilación e instalación de todas las partes del programa:

```
make all

make install
make install-init
make install-config
make install-commandmode
make install-webconf
```

Creamos una cuenta genérica para el acceso a la consola Web de Nagios, y reiniciamos el servicio *httpd* para aplicar todos estos cambios:

```
htpasswd –c /usr/local/nagios/etc/htpasswd.users nagiosadmin

service httpd restart
```

Ahora debemos realizar la instalación de los *plugins*, que es la configuración básica de Nagios por defecto:

```
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
make install
```

Un punto importantísimo en la instalación es añadir los servicios relacionados con Nagios al arranque, para que estos queden encendidos tras un reinicio del sistema:

```
chkconfig --add nagios
chkconfig nagios on
```

Para finalizar con la instalación de Nagios, verificamos que los ficheros de configuración de Nagios (definiciones de objetos) no tienen errores, y iniciamos el servicio:

```
nagios –v /usr/local/nagios/etc/nagios.cfg
service nagios start
```

Vamos a proceder a la instalación de Merlin. Merlin es el módulo que se va a encargar de convertir los fichero de configuración de objetos a un SGBD. Merlin se instala como servicio, y traduce automáticamente cualquier modificación que realicemos sobre los fichero .cfg de Nagios.

Se instalan las dependencias de Merlin:

```
yum install mysql-server libdbi-dbd-mysql libdbi-devel php-cli php-mysql
```

Creamos la base de datos para Merlin, y los permisos oportunos para su funcionamiento:

```
mysql -u root
mysql> create database merlin;
mysql> grant all privileges on merlin.* to merlin@localhost identified by 'merlin';
mysql> flush privileges;
mysql> quit;
```

Compilamos e instalamos Merlin. Es necesario instalar Merlin con los siguientes parámetros para que quede correctamente configurado:

```
make
./install-merlin.sh --nagios-cfg=/usr/local/nagios/etc/nagios.cfg
--dest-dir=/usr/local/nagios/addons/merlin
```

Llegados a este punto se realizan las comprobaciones del funcionamiento de Merlin. Reiniciamos los servicios *nagios* y *merlind* para comprobar que los objetos definidos en Nagios se añaden a la base de datos 'merlin'.

Tras comprobar por consultas sql que no funciona, editamos el script en php *object_importer_inc.php*, que erróneamente apunta a la cache de Nagios. Modificamos la ruta por */usr/local/nagios/var/objects.cache*.

Comprobamos que funciona correctamente. Seguimos con la instalación de Ninja.

El primer paso para la instalación de Ninja, es copiar los archivos de la aplicación a la ubicación deseada:

```
cp –a ninja /usr/local/nagios/addons/ninja
```

Se copia el fichero de configuración de http de la aplicación a la ruta y se edita modificando la referencia a la ubicación anterior:

```
cp op5build/ninja.httpd-conf /etc/httpd/conf.d/ninja.conf
```

Modificamos las referencias erróneas en los archivos *op5build/index.php, config/config.php* y *op5-upgradescripts/merlin-reports-db-updgrade.php*. Tras esto, se lanza el script de configuración de Ninja *install_scripts/ninja_db_init.sh* y se obtiene finalmente el siguiente resultado:

```
install_scripts/ninja_db_init.sh /usr/local/nagios/addons/ninja
Installing database tables for SLA report configuration
Upgrading SLA tables from v1 to v2 ... done.
Upgrading SLA tables from v2 to v3 ... done.
Upgrading SLA tables from v3 to v4 ... done.
Upgrading SLA tables to v5 ... done.
Installing database tables for AVAIL report configuration
Upgrading AVAIL tables from v1 to v2 ... done.
Upgrading AVAIL tables to v5 ... done.
Upgrading AVAIL tables to v6 ... done.
Installing scheduled summary reports
Database upgrade complete.
```

Reiniciamos el servicio *httpd* e intentamos acceder mediante un navegador Web a la aplicación:

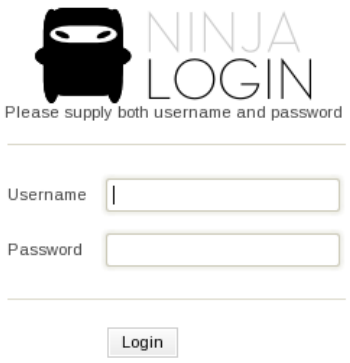

Ilustración 13 – Pantalla inicio aplicación Web Ninja

Finaliza así el primer paso de instalación del sistema base Nagios y Ninja.

Exportación de datos de MOM

Por regla general, el proceso de exportar los datos desde MOM a Nagios no va a ser automático. Como hemos dicho anteriormente, es necesario definir una estructura nueva, basada en la anterior pero con el objetivo de realizar un sistema eficiente sin excesivas reglas, grupos… que puedan saturar al operador.

Sin embargo, existe un elemento base que sí es posible migrar de forma automática; los *host* o servidores. El nuevo sistema debe monitorizar los servidores que anteriormente se monitorizaban, además de los nuevos servidores que se incorporan en la infraestructura. Qué mejor forma de asegurar que esto se cumple que extraer la información del propio MOM.

El mecanismo automático es el siguiente:

Utilizamos de nuevo SQL Server para consultar la lista de servidores que actualmente se monitorizan desde MOM:

```
SELECT Name
FROM dbo.Computer
WHERE ManagedType = 2
ORDER BY Name
```

Se programa una pequeña aplicación que dada una lista de equipos, crea los objetos Nagios para importarlos en el nuevo sistema.

```
StreamReader sr = new StreamReader("servidores.txt");
                  StreamWriter sw = new StreamWriter("hosts.txt");
                  string server = sr.ReadLine();
                  while(server != null)
                  {
                        Console.WriteLine(server);
                        try
                        {
                              IPAddress[] ips =
Dns.GetHostAddresses(server);
                              sw.WriteLine("define host{");
                              sw.WriteLine("\tuse\t\t\tWSERVIDORES");
                              sw.WriteLine("\thost_name\t\t" + server);

                              string adress = ("\taddress\t\t\t");
                              foreach(IPAddress ip in ips)
                              {
                                    adress = adress + ip.ToString() + ",
";
                              }
                              adress = adress.Substring(0, adress.Length
- 2);
                              sw.WriteLine(adress);
                              sw.WriteLine("\t}");
                        }
                        catch(Exception) {}
                        server = sr.ReadLine();
                  }
…
```

Todos los servidores que se monitorizan actualmente en MOM son servidores con sistemas operativos de Microsoft, por lo que se define una plantilla ALLWINSERVERS que nos permitirá definir configuraciones genéricas para este tipo de *hosts*.

Tras depurar los equipos obsoletos en MOM, ya disponemos de una monitorización básica de estos servidores.

Definición de objetos Nagios

La definición de los objetos *host* es la configuración base de nuestro sistema, la configuración final u objetivo es la definición de las reglas de monitorización

(*services* en Nagios). Pero antes de alcanzar ese punto, es necesario definir la estructura que organizará de forma eficiente Nagios y que ayudará a los operadores a la supervisión del sistema.

Los objetos *services* no deberían aplicarse sobre *host*, ya que en una infraestructura tan compleja, varios *host* implementan los mismos servicios y esta relación es susceptible de cambios en los objetos. Es por ello que se va a definir una estructura jerarquizada de *hostgroups* y *servicegroups* basada en los servicios que se implementan sobre el cliente, y en la distribución geográfica de los servidores.

La definición de *hostgroups* y *servicegroups* es posiblemente la más importante dentro del este sistema. En una infraestructura tan extensa (más de 100 servidores) es necesario mantener un sistema bien estructurado con el objetivo de mejorar la supervisión del mismo, y además de dotar de una filosofía de implementación para los futuros administradores.

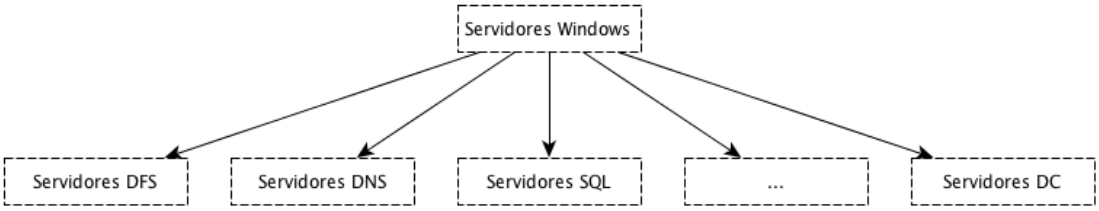

Ilustración 14 – Hostgroups basados en servicios del Catálogo

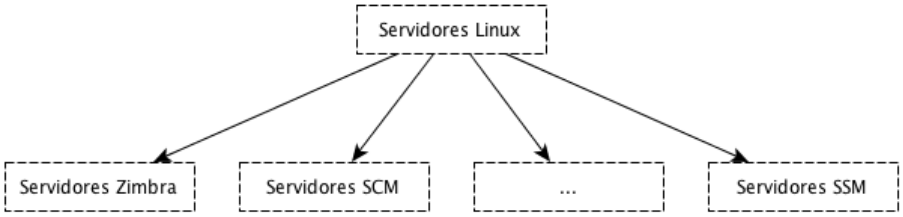

Ilustración 15 – Hostgroups basados en servicios del Catálogo

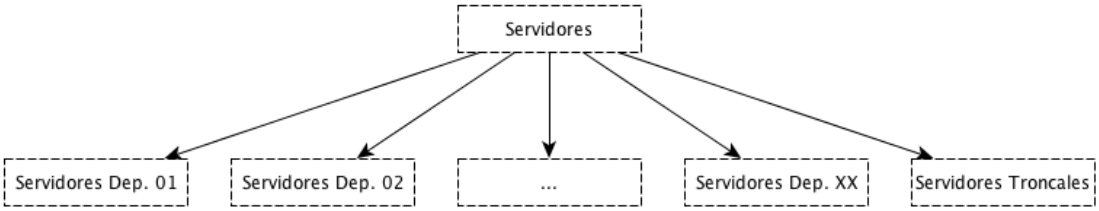

Ilustración 16 – Hostgroups basados en distribución geográfica

Vamos a dar algunos ejemplos de la utilidad de disponer de un sistema de grupos correctamente jerarquizado:

- “Servidores Windows” y “Linux” permiten definir *services* de monitorización a nivel de sistema operativo y hardware.
- “Servidores DFS”, “DC”, “DNS”… permiten definir *services* según los roles que implemente cada servidor.
- “Servidores Dep. 01” “Dep. 02”… permiten definir configuraciones particulares para cada departamento (contactos del departamento, *timeperiods* para el departamento…).

Además, tal y como hemos dicho, la distribución por grupos ayuda a presentar la información al operador de un modo más ordenado.

La definición de *servicegroups* pretende imitar el elemento de *Rule Group* de MOM. La disponibilidad de un determinado servicio del catálogo se calcula en base al estado de un grupo de *services* que se encargan de su monitorización.

Como hemos dicho anteriormente, la abstracción de las tecnologías de los servicios del Catálogo de Servicios ofrecidos al cliente es un requerimiento de ITIL. Este requerimiento pretende asegurar que tras un cambio tecnológico, el grado de disponibilidad y capacidad no se ven afectados.

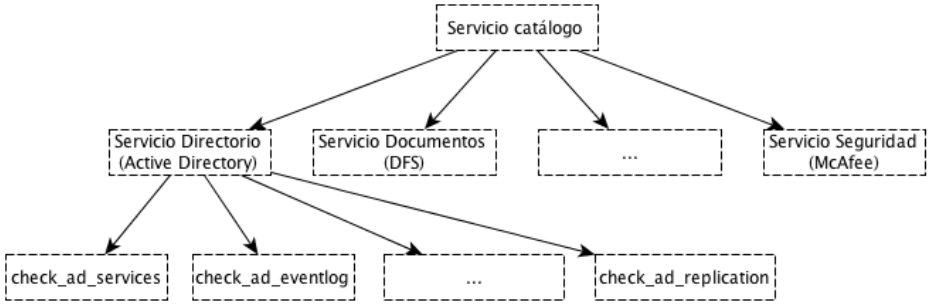

Ilustración 17 – Servicegroups basados en servicios del Catálogo

La definición de los *hostgroups* y *servicegroups* se realiza de forma manual. Los *hostgroups* se pueden definir en base a las configuraciones de MOM y al conocimiento de la infraestructura. Los *servicegroups* podemos definirlos, pero se relacionarán con los distintos *services* a medida que estos sean implementados en el siguiente punto.

Otros objetos a definir importantes para el funcionamiento normal del sistema son los *contacts*, *contactsgroups* y *timeperiods*. Los *contacts* y *contactgroups* se definen con los datos extraídos de los operadores y grupos de MOM. La información más importante de un operador en MOM son sus datos de contacto (mail, teléfono…) y los periodos en los que deben recibir notificaciones.

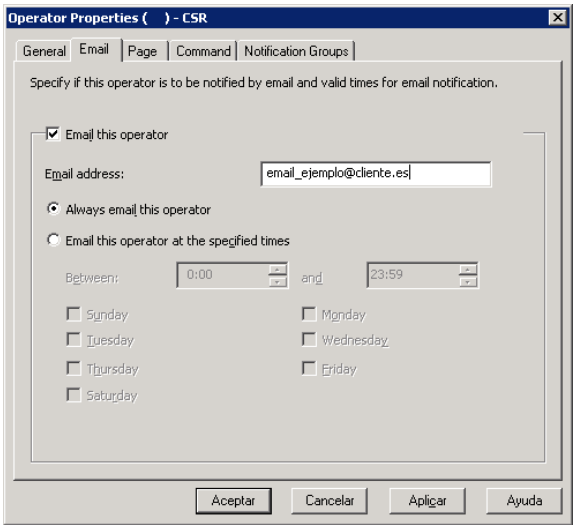

Ilustración 18 – Contacto en MOM

Basándonos en estos periodos definimos además los *timeperiods*, que marcan los horarios de notificación para cada contacto. Para este cliente todos los contactos tienen asociado el *timeperiod* 24x7, por lo que no resulta complicado la definición de estos objetos.

Programación de scripts

Como ya se ha dicho, replicar y configurar los eventos de monitorización de los servicios implementados correctamente es la tarea que más tiempo puede consumir dentro de este proyecto. Es por ello, que se opta por cubrir la monitorización básica de los servicios, y posteriormente corregir las deficiencias del sistema o incorporar nuevos eventos y scripts que apoyen la monitorización.

Mas del 75% de las reglas actualmente habilitadas en MOM se basan en la supervisión del estado de los eventos del visor de sucesos de los sistemas Windows. Vamos a centrarnos en programar una aplicación sencilla de consulta de eventos por WMI, y en extraer tablas de eventos basadas en los errores críticos de los Management Packs de MOM.

La aplicación realiza la consulta que hemos nombrado anteriormente, y esta se basa en los ficheros CSV creados a partir de los datos de los Management Packs de MOM:

```
1547;NTDS General;40;A schema collision occurred while replicating a class definition
1546;NTDS General;40;A schema collision occurred while replicating attribute definitions
1165;NTDS Replication;40;Active Directory cannot allocate a new domain controller GUID
1016;NTDS General;40;Active Directory cannot be initialized
```

Además, el trabajo de este script se apoya con la monitorización de servicios desde el lado servidor, y de la incorporación de otros scripts disponibles en el repositorio de scripts de Nagios. Por seguridad, estos últimos deben disponer de código fuente, comprobarse debidamente y verificar que su comportamiento no altera el funcionamiento de los servidores.

Instalación de NRPE y distribución masiva de scripts

Se ha optado por no utilizar ningún software de distribución para la instalación del servicio NRPE en los servidores a monitorizar. Es necesario conocer de primera mano el resultado de la instalación, y si realmente los resultados que el servicio NRPE manda al servidor de monitorización son correctos.

La instalación de NRPE como servicio en Windows es sencilla y se lanza por línea de comandos mediante la instrucción:

```
nrpe_nt -i
```

La configuración del servicio se realiza mediante un fichero de configuración .cfg. En este fichero se definen los distintos *services* que pueden lanzarse sobre este servidor ejecutando los scripts ubicados en el mismo.

Tras esto se va a programar una aplicación para distribuir los scripts de monitorización y actualizar la configuración NRPE remotamente. Esta aplicación realizará lo siguiente:

1. Extraer lista de equipos a distribuir de los *hostgroups* definidos.
2. En cada uno de ellos:
    a. Detiene el servicio NRPE
    b. Modifica el fichero de configuración .cfg
    c. Copia los scripts necesarios a la carpeta *libexec*
    d. Reinicia el servicio NRPE
3. Define en el fichero de configuración de *services* el nuevo *service* aplicado sobre el grupo *hostgroup* del primer paso.

Definición de pruebas y control de funcionamiento

Tras la configuración de los objetos de Nagios esenciales, y de la monitorización del lado cliente, queda por delante una etapa de inspección y reconfiguración del sistema objetivo. En este periodo, los técnicos del proyecto deben supervisar las alertas recibidas desde MOM y las recibidas desde Nagios, para corregir posibles problemas:

- **Alertas inútiles:** alertas que no son útiles para la detección de problemas, que aportan información poco clara y/o saturan el sistema de alertas.
- **Alertas erróneas:** alertas que devuelven un estado de *Error* o *Warning* cuando el servicio realmente está correcto.

El periodo de convivencia entre ambos sistemas de monitorización debe ayudar a definir un sistema objetivo mucho más realista. Incluso tras deshabilitar el funcionamiento de MOM (que consiste en deshabilitar el envío de alertas a todos los operadores) el equipo técnico del proyecto puede continuar monitorizando desde MOM con el fin de detectar posibles problemas en los que no se pensó en un primer momento.

Hay que ser conscientes que los problemas en los sistemas monitorizados son aleatorios y pueden no aparecer durante muchos meses, años, incluso nunca. Dentro de una infraestructura extensa como la que tiene el cliente objeto de

estudio hay más probabilidades de encontrar problemas inéditos para el operador durante este periodo de convivencia.

Tras la desinstalación de MOM, es necesario mantener un trabajo de reconfiguración constante que permita crecer al nuevo sistema de monitorización. Ante un problema que afecte a la disponibilidad o capacidad del servicio y que no haya sido detectado por el sistema de monitorización, el equipo de resolución debe añadir a las tareas de investigación de la causa del problema la tarea de documentar las variables que se dan como aviso ante la incidencia.

Por ejemplo, un problema de corrupción de las zonas DNS ocasionan una incidencia grave de validación en Active Directory. El equipo de resolución en las tareas de investigación debe detectar:

- **Avisos previos al problema:** métricas que pueden ayudarnos a predecir que este problema se va a producir.
- **Avisos en el problema:** métricas que nos avisan que nuestro sistema está comprometido.

Las tareas de investigación deben detectar y definir nuevos eventos en base a:

- **Situación del hardware:** discos llenos, procesadores demasiado ocupados…
- **Situación del sistema operativo:** fallo en librerías del sistema, fallo en aplicaciones instaladas…
- **Visor de eventos:** ID's detectados en cada momento en los Event Logs del servidor.

Con estos datos los administradores del sistema de monitorización pueden definir nuevos eventos que impidan que se repita de nuevo una situación de indisponibilidad similar.

Conclusiones

El estudio actual pretende migrar lo monitorizado por MOM a Nagios. Existen varias forma de afrontar e interpretar esta migración, y cada una de ellas tiene sus pros y contras.

Podemos considerar la migración como una configuración desde cero, creando los eventos y las configuraciones de alertas desde un punto inicial. La principal ventaja de esto es transformar radicalmente el sistema de monitorización, eliminando los excesos que puede tener hoy en día un sistema tan complejo como MOM.

Pero debemos tener en cuenta que será necesario un periodo más largo para poseer un sistema de monitorización en condiciones, ya que su desarrollo está basado en las necesidades que se van encontrando día a día en la infraestructura. Además, nadie asegura que con este desarrollo no se llegue otra vez a un sistema complejo. La inclusión y rectificación de alertas debe basarse en una filosofía de diseño clara y concreta.

Otra posibilidad es tomar como referencia todo lo que MOM monitoriza actualmente e intentar migrarlo a Nagios. Es necesario realizar un paso intermedio que permita depurar y filtrar eventos y alertas innecesarias, saber como implementar ciertos aspectos útiles en MOM, como por ejemplo, los criterios de registro de eventos (*Override Criteria*).

La ventaja de esta migración es poseer de un sistema de monitorización similar al anterior. Al tratarse de una implementación similar, los operadores no tienen tanta dificultad para migrar la forma de trabajo a Nagios. La principal ventaja es también su principal inconveniente; podemos estar migrando eventos y reglas inútiles o que simplemente dentro del entorno de Nagios no funcionan todo lo bien que lo hacían en MOM.

Conclusiones

Trabajo desarrollado

El presente estudio es un trabajo de investigación de dos tecnologías de monitorización. Aunque su principal objetivo es ofrecer la información necesaria para poder efectuar la migración, se ha alcanzado un nivel de conocimiento alto tanto de la actual herramienta como de la herramienta objetivo.

Antes de abordar el estudio de MOM, la información o la visibilidad que se tenía de la herramienta es que esta era totalmente cerrada e imposible de desentrañar. Tras el estudio hemos podido ver como MOM se basa mayoritariamente en elementos no tan desconocidos, como son los scripts VBS y los Event Log de cualquier sistema Windows.

También se ha estudiado a fondo el sistema objetivo Nagios. La idea principal de este estudio es conocer las posibilidades que ofrece Nagios, para configurar de la mejor forma y mediante sus objetos una herramienta de monitorización equiparable a la actual. Teniendo bien claro este diseño estructural de *hosts*, *hostgroups*, *services*... podremos mantener un crecimiento correcto del sistema que permita a cualquier administrador corregir y añadir nuevas soluciones.

A lo largo del documento hemos hecho énfasis en la importancia de ITIL, sobre todo de los procesos de Disponibilidad y Capacidad del servicio. Estos procesos van a ayudar a diseñar una infraestructura fiable y rentable, basándose en los resultados que obtendremos de la herramienta de monitorización.

Aportaciones

El presente documento realiza varias aportaciones aplicables a otros clientes y a otras empresas de tecnologías de la información.

Estudio del sistema de monitorización MOM

El sistema de monitorización MOM es una herramienta propietaria compleja de la cual se desconoce frecuentemente su funcionamiento interno.

La investigación realizada en este documento puede ser de gran utilidad no solo para otras migraciones tanto a Nagios u otras herramientas de monitorización, sino también para:

- *Parametrización* correcta de la herramienta MOM en otros clientes.
- Resolución de incidencias y problemas internos de MOM en otros clientes.

Diseño de un sistema de monitorización basado en Nagios

Son varios los aspectos a destacar en el diseño de la herramienta Nagios que se ha realizado en este documento.

La propuesta estructurada de grupos de objetos puede ser útil como punto de partida para cualquier organización que necesite realizar un nuevo diseño. Por otra parte, la utilización de la información de MOM para la migración a Nagios (script Event Log basado en Management Packs, scripts de migración…) son elementos reutilizables en cualquier otra migración entre ambos sistemas.

Estudio de migración del sistema

La documentación de instalación de Nagios y Ninja resulta de gran utilidad, ya que como hemos podido ver ha sido necesario introducir alguna modificación en los procedimientos oficiales para compatibilizar las herramientas en Red Hat Enterprise Linux 5.5.

Por otro lado, el proceso de extracción y documentación de los XML de los Management Pack es aplicable a cualquier otro implementado en otros clientes.

Alineación de la monitorización a las prácticas ITIL

La homologación de ITIL con Nagios es una aportación importante. La monitorización de SLA's mediante la interfaz gráfica Ninja ayuda a la extracción de reportes útiles para los procesos de ITIL.

Además, el diseño de la estructura de Nagios tiene como fin satisfacer los requerimientos de ITIL:

- Abstracción de tecnologías en servicios del Catálogo de Servicios.
- Monitorización en base a la disponibilidad y capacidad del servicio.

Ampliaciones del proyecto

Conectividad entre servidores: puertos, firewall…

Un aspecto muy concreto del cliente objeto de estudio es su política de seguridad, que por defecto bloquea cualquier puerto hacia cualquier otro servidor y desde cualquier otro servidor. Esta política tan útil en lo que a seguridad se refiere, resulta una dificultad añadida al proyecto de migración.

Es necesario contar con este aspecto a la hora de realizar el laboratorio previo al proyecto. Se debe contar con los puertos necesarios para el funcionamiento correcto tanto del sistema operativo como de la aplicación Nagios.

Si la solicitud de puertos se realiza a un proveedor distinto a la organización TI y al cliente, será necesario definir un protocolo de petición correcto, y evitar en la medida de lo posible realizar peticiones antes de disponer de una lista final de los puertos necesarios.

Por otra parte hay que tener en cuenta la configuración firewall tanto del servidor encargado de la monitorización, como de los servidores monitorizados.

Al no disponer de un entorno gráfico tanto en el servidor Linux de Nagios como en los servidores Linux de los demás servicios, será necesario definir reglas en las *iptables* de los sistemas.

Modificaciones de Nagios: mantenimiento centralizado de NRPE

Como hemos comentado en este documento, con la migración a Nagios se pierde claramente capacidad de administración centralizada del sistema de monitorización. MOM disponía de un sistema propio para la distribución de scripts, definiciones… de forma transparente para el administrador.

Desde Nagios no es posible mantener de forma centralizada los archivos necesarios para la monitorización desde el lado cliente. Esto en clientes pequeños, con cinco a diez servidores puede resultar trivial, pero no lo es para una infraestructura de más de cien sistemas.

La solución pasa por implementar un módulo en el servidor de monitorización que mantenga actualizado el sistema distribuido de monitorización. Se debe distribuir a los clientes los datos actualizados del servidor de forma transparente o semitransparente para el administrador.

Puede implementarse un mecanismo de gestión de versiones simple en el servidor de monitorización, que compruebe las fechas y/o el *checksum* de los archivos necesarios para mantenerlos actualizados en todo momento.

Entrega del servicio: documentos de diseño y explotación de la herramienta

En el apartado de introducción teórica de este proyecto hemos visto como ITIL consiste en un ciclo de vida circular, y que los procesos de Gestión de la Disponibilidad y la Capacidad, y en consecuencia el proyecto de diseño de la nueva herramienta de monitorización se englobaban en la etapa de Diseño del Servicio.

Una norma fundamental de ITIL es el traspaso de conocimiento entre las distintas etapas. El objetivo es que nuestro sistema de monitorización desarrollado en la etapa de Diseño del Servicio, llegue a la etapa de Operación del Servicio mediante la etapa intermedia de Transición del Servicio.

En la etapa de Transición se deben redactar los documentos necesarios para que a corto plazo la herramienta de monitorización se explote en la fase de Operación. Los documentos esenciales para ello son:

- **Documento de diseño o configuración:** debe ofrecer una visión global de cómo se ha diseñado la herramienta. No debe incluir información de la migración ni del sistema anterior, y se debe hablar de la definición del sistema objetivo, no de su funcionamiento interno.
- **Documento de operación o explotación:** debe explicar al operador cómo debe utilizar la herramienta desde el punto de vista del diseño.

Los documentos de la fase de Transición suelen ser redactados por los técnicos del proyecto, ya que estos son los que conocen mejor la nueva herramienta, su configuración inicial y cual es su funcionamiento deseado.

Se debe tener en cuenta que estas guías deben de ser muchas veces sencillas y prácticas, destinadas a cualquier tipo de operador que podamos encontrar en la organización IT.

Referencias

Bibliográficas

[1] Randy Holloway, Telmo Sampaio, Russ Kaufmann, Marcus Oh, Derek Comingore. *Professional MOM 2005, SMS 2003 and WSUS*. Wrox 2007
[2] Office Of Government Commerce. *Service Design*. London. TSO 2007

Internet

[3] Comunidad de Nagios. http://nagios.sourceforge.net/docs/3_0/. *Documentación Nagios 3.0*
[4] Microsoft. http://technet.microsoft.com/. *TechNet Support*
[5] Osiatis. http://itil.osiatis.es/. *ITIL v3 – Gestión de Servicios TI*